

Governors State University OPUS Open Portal to University Scholarship

All Capstone Projects

Student Capstone Projects

Summer 2011

Online Assistance

Imran Khan

Governors State University

Follow this and additional works at: <http://opus.govst.edu/capstones>



Part of the [Software Engineering Commons](#), and the [Systems Architecture Commons](#)

Recommended Citation

Khan, Imran, "Online Assistance" (2011). *All Capstone Projects*. 41.
<http://opus.govst.edu/capstones/41>

For more information about the academic degree, extended learning, and certificate programs of Governors State University, go to
http://www.govst.edu/Academics/Degree_Programs_and_Certifications/

Visit the [Governors State Computer Science Department](#)

This Project Summary is brought to you for free and open access by the Student Capstone Projects at OPUS Open Portal to University Scholarship. It has been accepted for inclusion in All Capstone Projects by an authorized administrator of OPUS Open Portal to University Scholarship. For more information, please contact opus@govst.edu.

A PROJECT REPORT ON

ONLINE ASSISTANCE

Submitted to COLLEGE OF ART AND SCIENCES TO
GOVERNORS STATE UNIVERSITY

Award of Degree for

MASTERS

IN

COMPUTER SCIENCE

DONE BY

NAME: IMRAN KHAN



ONLINE ASSISTANCE

1. INTRODUCTION

1.1 COMPANY PROFILE

1.2 INTRODUCTION TO PROJECT

1.3 HARDWARE AND SOFTWARE

REQUIREMENTS

2. SYSTEM ANALYSIS

2.1 INTRODUCTION

2.2 FEASIBILITY STUDY

2.3 DATACOLLECTION

3. SYSTEM DESIGN

3.1 INTRODUCTION

3.2 SOFTWARE TECHNIQUES

3.2.1 MODULARITY

3.2.2 MODULAR INTERACTION

3.3 DATA DICTIONARY

3.4 DATA FLOW DIAGRAMS

3.5 E-R DIAGRAM

4. SYSTEM DEVELOPMENT

4.1 CONCEPTS AND TECHNIQUES

5. TESTING AND IMPLEMENTATION

5.1 INTRODUCTION

5.2 IMPLEMENTATION

6. SCREENS

7. CONCLUSION

7.1 FUTURE ENHANCEMENT

7.2 CONCLUSION

8. BIBLIOGRAPHY

INTRODUCTION

1.1 COMPANY PROFILE

// Paste Company Profile Here

INTRODUCTION TO PROJECT

- Simple and intuitive list views of tickets with paging, sorting, and filtering features to quickly help you locate tickets you are interested in without having to create and manage custom "queries" for common needs.
- A full featured search page for times when the built in lists just don't do the trick. You can search title, details, or tags and there are a variety of filtering options to help narrow your search further.
- A simplified ticket creation screen with a minimum of required fields. Users won't feel like they are answering to the Spanish inquisition when they need to report a new problem and they won't have to answer a lot of technical questions they don't understand. Mostly, they just pop in a title, add some details, then hit create button.
- Ticket viewer that displays information about a ticket simply and effectively.

- An action oriented ticket system that intuitively guides the users through a simple, but effective workflow. Each action can be accompanied by comments to assist in communicating the ticket's progress as it is worked through to completion.
- Running activity log that tracks everything done to a ticket, who did it, when, and why. This is the meat-and-potatoes of a ticket and acts as both discussion board and audit trail.
- A streamlined workflow that allows tickets to move effortlessly through the process from "active" to "closed" with no unnecessary obstacles or complex "approval" and "review" processes. The workflow emphasizes efficient communication and cooperation between users and help desk staff.
- Email notifications for changes to tickets.
- Offers a self-contained security database for users and roles, or it can integrate with Windows security and your Active Directory domain.

1.3 HARDWARE AND SOFTWARE REQUIREMENTS

Hardware Specifications:

The following are the minimal hardware requirements to execute this Project:

- Intel P-III Processor or later
- 256 MB RAM
- 2 GB Hard Disc space minimum

Software Specifications:

Operating System : Windows NT/ Windows XP with IIS Server.

Language : ASP.NET, AJAX, C#.

IDE : Microsoft Visual Studio.Net 2005 / 2008.

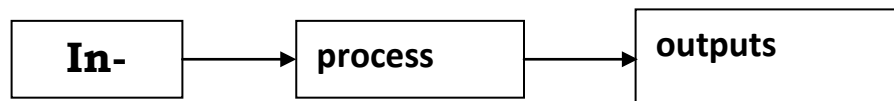
Backend : SQL Server 2005

2. SYSTEM ANALYSIS

2.1 INTRODUCTION

This project entitled Online Assistance is efficient and designed to do only one thing, facilitate communications between help desk staff and end users. The overriding design goal is to be as simple and frictionless for both users and help desk staff as is possible. It can leverage SQL server for membership and role based security or integrate with windows authentication and Active Directory groups.

Software Engineering occurs as a consequence of a process called "System Engineering". Instead of concentrate solely on software, system-engineering focuses on a variety of elements like Analyzing, Designing and organizing these elements into a system.



In System Analysis and Design the concern is usually with the man made systems involving inputs, processes and outputs.

System:

A set of interactive elements responding to inputs to produce outputs. The System Engineering process is called business process engineering when the context of the engineering work focuses on a business enterprise. The goal may be to support same business function that can be sold to generate business revenue. To accomplish the goal, a computer-based system makes use of a variety of system elements.

Software:

Computer programs, data structures and related documentation that serve to affect the logical method, procedure or control that are required.

Hardware:

Electronic devices that provide computing capability, interconnectivity devices that enable the flow of data and Electro-mechanical devices that provide external world functions.

People:

Users and operators of both hardware and software.

Database:

A large organized collection of information that is accessed via software.

Documentation:

Descriptive information that portrays the use and/or operation of the system.

Procedures:

The steps that defines the specific use of each system element or procedural context in which the system resides.

SYSTEM ANALYSIS:-

System Analysis is the process of analyzing a system with the potential goal of improving or modifying the system. Analysis is breaking down the problem into smaller elements for study and ultimately providing better solution.

During the process of system development analysis is an important aspect.

This involves gathering and interpreting facts, diagnosing the problem and using the information to recommend improvements to the system. Ultimately, the goal is to give a computerized solution.

The system considered was analyzed holding in view the relevant facts pertaining to the Online Assistance. And its various activities, stressing emphasis on gathering information regarding its requirements and specifications.

Requirement Analysis:

It provides the appropriate mechanism for understanding what the customer wants, analyzing need, assessing feasibility, negotiating the reasonable solution, specifying the solution un-

ambiguously, validating the specifications and managing the requirements as they are transferred into an operational system.

As the requirements analysis activity commences the following are asked and answered:

- ❖ Is each requirement consistent with the overall objective for the system?
- ❖ Have all requirements specified at the proper level of abstraction?
- ❖ Is the requirement really necessary or does it represent an add-on feature that may not be essential to the objective of the system?
- ❖ Is requirement bounded and unambiguous?
- ❖ Does each requirement have attribution?
- ❖ Do any requirements conflict with other requirements?
- ❖ Is each requirement achievable in the technical environment that will house the system
- ❖ Is each requirement testable once implemented?

FEASIBILITY STUDY

After Requirement Analysis we check for whether the computerization of this Online Assistance is feasible or not.

.System Performance Definition:

Statement of constraints

Constraints are factors that limit the solution of the problem. Some constraints are identified during the initial investigation and are discussed with the user. There are general constraints that might have a bearing on the required performance of a candidate system.

Identification of specific system objectives.

Once the constraints are spelled out, the analyst proceeds to identify

The system's specific performance objectives. The anticipated benefits are

- 1) Improved collection schedule.
- 2) Cost reduction.
- 3) Physical space reduction.
- 4) Improved customer service.

Analysis Principles:

- 1) The information domain of a problem must be represented and understood.
- 2) The functions that the software is to perform must be defined.
- 3) The behavior of the software must be represented.
- 4) The models that depict information function and behavior must be partitioned in a manner that uncovers details in layered fashion.
- 5) The analysis process should move from essential information toward implementation detail.

By applying these principles, the analyst approaches a problem systematically. We have to understand the problem clearly before we begin to create the analysis model.

Specifications:

There is no doubt that mode of specification has much to do with the quality of the solution. Software Engineers who have been forced to

work with incomplete, inconsistent or misleading specifications have experienced the frustration and confusion that variably results.

"So the specifications must be tolerant of incompleteness and augmentable".

The Software Requirement Specifications:

The Software Requirement Specification is produced at the culmination of the analysis task.

Some proposed candidate formats for Software Requirement Specification:

- ❖ The Introduction of the Software Requirement Specification states the goals and objectives of the software.
- ❖ The Information Description provides detailed description of the problem that the software must solve.
- ❖ A description of each function required to solve the problem is presented in the "Functional Description".
- ❖ Validation Criteria is probably the most important and ironically, the most often neglected section of the Software Requirement Specification.

Both the software engineer and the customer conduct a review of Software Requirement Specification. Because the specification forms the foundation of the development phase, extreme care should be taken in conducting the review.

User Requirement Specifications:

The user requirements and specifications to be documented without any ambiguity. The information regarding activities involved and data was gathered by meeting the people. This involved a wide range of discussions covering all the aspects of requirements. One important aspect that is observed after the commencement of the

system. Development is the major change in user requirement specifications. Which have to handle without major alterations of software system.

The requirement features include:

- ❖ User interface form for navigating and selecting a particular option.
- ❖ User interface forms with insertion, deletions and modification.
- ❖ Report generation required format of reports has to be included.

System Requirement Specification:

Taking the user requirement into consideration an understanding of the system requirement can be comprehended.

The system should have the following features bundled:

- ❖ Database support
- ❖ Reports
- ❖ Flexibility
- ❖ Menu bar
- ❖ Dropdown Menus

Software and Hardware Selections :

Hardware Selection:

In this we need Personal Computer, a Floppy Disk, and a Printers are required for this Project. These requirements are to be met by the system. Hence the Project is technically Feasible.

Software Selection:

This Project requires a language, which supports a Personal Computer on Client/Server development environment. Microsoft Sql Server is required as backend and ASP.NET, AJAX as front End Tool and c#.NET for business logic.

Economic Feasibility:

Taking cost benefit analysis into consideration the benefit of the proposed system surpasses the cost. Further the cost of the software and hardware is nullified. Since the system is developed taking into consideration the available resources in the Company.

Technical Feasibility:

The system has no constraints regarding to the availability of the software and hardware .The system has capability to hold the required amount of data taking into consider the system

Operational Feasibility:

There is a great deal of support and encouragement for the system from the company. End users are aquatinted with the operating environment and need a briefing of its operation. The manual will solve the problems of the software.

Existing System

At my employer's office we use a help-desk application called Census by MetaQuest software. Census, to put it mildly, is total crap. It is slow, cumbersome, complex, and annoying to use. It is also a nightmare to install, keep running, and administer.

When searching for alternatives we came to realize that nearly all of the help desk applications on the market suffer from the same kinds of problems.

They all have these complex customizable workflows you have to manage, complex permission systems, and so many fields that even the dedicated IT staff end up ignoring most of them. They Al-

so tend to be hard to install, difficult to configure, and tedious to maintain.

Then you have the endless categorization... what's the priority, what's the severity, what's the impact on the users, what kind of problem is it, what kind of user reported the problem, what operating system is installed on the system with the problem, what gender was your first pet dog....

ENOUGH!

The end-user doesn't want complexity.

Requiring end-users to fill out complex forms with all kinds of technical jargon just scares them... which means they don't fill out tickets until their small problems have become really big ones.

The help desk doesn't want complexity either.

Sure, they are used to complexity, but our help desks is over-worked, under-staffed, and are under a lot of pressure to fix everything RIGHT NOW!

The marketing for most help-desk applications will go on about search ability, knowledge-bases, and how your help desk tickets can be used to build a database of problems and solutions which will improve efficiency by leveraging past experiences. Some even promise to turn your issue tracker into a customer-facing self-help support web site (if you aren't ready to vomit now, you haven't worked on a real help desk before).

That's a great idea, and it does well the sales meetings with upper management.

It overlooks the obvious fact that a help-desk system is exactly the wrong source for that kind of knowledge sharing. End users describe their problems as best as they can, which means they describe their problems badly at best. The help desk staff will do an equally poor job explaining what they did and how they fixed the problem.

That's just how it is.

No amount of policy, management directives, training, or threats will change the fact that data in our project will be hard to search, sketchy about the details, and will totally fail to make a useful knowledge base.

To be successful, the system needs to be efficient, fast, and easy to use. It needs to allow for quick and frictionless communications between the users and IT staff. And most importantly, the system must NOT get in the way. It must be nearly effortless to use and so simple that even a Mac user could figure it out.

Proposed System:

Online Assistance was "planned". It was just something we decided to in my spare time. It needed to support the end-users and IT staff, but without any unnecessary administration or maintenance.

So the general idea was to implement the simplest Online Assistance was "planned". It needed to support the end-users and IT staff, but without any unnecessary administration or maintenance.

So the general idea was to implement the simplest Online Assistance as possible. It would have a very simple non-customizable workflow, as few data fields as possible, and be easy to setup, maintain, and administer.

The other goal was to use as little code as possible. So Online Assistance would have to leverage everything the .net 3.5 framework could muster that might reduce complexity and improve my productivity.

In formulating the basic idea, we also realized that Online Assistance could probably meet the needs of most small IT Companies, and probably many larger ones as well.

Online Assistance Design:

Online Assistance is designed as a pure help desk issue tracking system. The focus is on the relationship between your users and your IT staff.

It doesn't attempt to be a customer facing support product.

It doesn't attempt to be a customer relationship management system.

And it is not a bug-tracking system to support the software development life cycle.

Online Assistance allows users to submit tickets. It provides help desk with a way to keep track of the tickets. And it facilitates smooth communication between users and help desk.

Online Assistance concentrates on being simple and effortless. It has very few data entry fields, and demands almost no configuration or administration overhead.

Online Assistance leverages several common web 2.0 concepts to make tickets easier to track and to make working through a ticket more like having a conversation.

Online Assistance probably isn't for everyone though. Large shops that demand tight administrative control over process and permissions will certainly not like it. It would have a very simple non-customizable workflow, as few data fields as possible, and be easy to setup, maintain, and administer.

The other goal was to use as little code as possible. So Online Assistance would have to leverage everything the .net 3.5 framework could muster that might reduce complexity and improve my productivity.

In formulating the basic idea, we also realized that Online Assistance could probably meet the needs of most small IT companies, and probably many larger ones as well.

Online Assistance is designed as a pure help desk issue tracking system. The focus is on the relationship between users and IT staff.

It doesn't attempt to be a customer facing support product. It doesn't attempt to be a customer relationship management system. And it is not a bug-tracking system to support the software development life cycle.

Online Assistance allows users to submit tickets. It provides help desk with a way to keep track of the tickets. And it facilitates smooth communication between users and help desk.

Online Assistance probably isn't for everyone though. Large shops that demand tight administrative control over process and permissions will certainly not like it.

Objective for the proposed system

- ❖ To give quick access to information.
- ❖ To decrease the communication gaps and increase the coordination.
- ❖ To build a database of information regarding the transactions.
- ❖ For decision-making and strategy planning.

Description of the proposed system:

- ❖ Graphical User Interface
- ❖ The System is application for Querying, Updating the information.
- ❖ A user-friendly interface compliant to Windows has to be created.
- ❖ This includes menus, drop down menus with all options like
- ❖ Insertion, deletion and modification.

DATA COLLECTION

Some of the customers don't have the time to attend the interviews so questioners are prepaid on company and they are passed

to the employees in the time to they used to response to the questioners. It is another best method to gather the data.

- As soon as the client enters the system, they have to log into the project or register as a new client to avail the services.
- Once the clients provides the authorization information to the system, it presents with the client's area called main page based on the type of the user (by default for newly registered users project assigns the ticket submitter's role).

Introduction

Once software requirements have been analyzed and specified, software design as the first of 3 technical activities – design, code generation and test-that are required to build and software.

Stages of Design of a Project are:

- a. Conceptual Design
- b. Database Design
- c. Functional Design

1) Conceptual Design:

The conceptual structure of a Database is called schema. Schema shows the kind of data that exists in a database and how these are logically related to each other. A schema can be regarded as a blueprint that portrays, both, kind of data used in building a database and logical relationship, and must correctly represent their interrelationships. Schema is frequently depicted pictorially viz., Entity Relationship Diagrams (E-R Diagram), Data Flow Diagram (DFD) and UML diagrams.

2) Database Design:

Note: Owing to the confidential nature of the system, the complete database details could not be included in the system.

DBMS: A Database is a stored collection of interrelated data, organized on the basis of relationship in the data rather than the convenience of storage structures. It enables sharing of data among various users as and when required. Database Management System is software that provides more flexibility in the storage and retrieval of data and productive information.

Relational Database: A database, in which data is stored in tables, allowing relationship among tables and more efficient non-redundant data storage and manipulation.

Table: Table is a fundamental structure of Relational Database Management System. In tables data is stored in rows (records) and columns (fields). The data is usually about a particular category of things, such as employees or books etc.,

Field: An element of a table that contains a specific item of information such as Account Code. A field is represented as a column.

Relationship: An association established between common fields in two tables. A relationship can be one-to-one (or) one-to-many (or) many-to-many.

Record: A collection of data about a particular thing like a book. A record is represented as a row in a table.

Primary Key: Collection of one (or) more fields whose value/values uniquely identifies a record in a table. This key is used to identify, each occurrence of an entity. It never has a null value.

Foreign Key: One or more table fields that refer to the Primary Key field/fields of another table. A foreign key is used to create links with another entity.

Security: A set of features used to specify (or) restrict access to specified users (or) groups. Giving Passwords and Usernames to access the database may impose security.

3) Functional Design:

Input describes the information to be supplied to this function either by the user on the screen (or) from any data store.

Processing describes the operations to be carried out by the function. The validations necessary are also described at this point.

Output describes the information obtained from the function (or) the action carried out by the function.

Design principles:

- Software designs is a both process and a model Basic
 - Design principles enable the analyst to navigate the design process.
- The design process should not suffer from “tunnel vision”.
- The design should be traceable to analysis model.
- The design should “minimize the intellectual distance” between software and problem that exists in the real world.
- The design should exhibit uniformity and integration.
- The design should be structured to accommodate.
- The design should be structured to degrade gently, even when aberrant data, events or operating conditions are encountered.
- The design should be reviewed to minimize conceptual (semantic) errors.

Input Design:-

Inaccurate input data are the most common cause of errors in data processing. Errors entered by data entry operator can be controlled by input design.

Input design is the process of converting user-originated inputs to a computer-based format. Once defined, appropriate input media are selected for processing. The goal of the designing input data is to make data entry as easy, logical and free from errors as possible.

Output Design: -

Computers output is the most important and direct source of information's to the user. Efficient, intelligible output design should improve the systems relationships with the user, and help in decision-making.

SOFTWARE TECHNIQUES

SOFTWARE ENGINEERING PARADIGM APPLIED (RAD-MODEL)

The two design objectives continuously sought by developers are reliability and maintenance.

Reliable System

There are two levels of reliability. The first is meeting the right requirements. A careful and through systems study is needed to satisfy this aspect of reliability. The second level of systems reliability involves the actual working delivered to the user. At this level, the systems reliability is interwoven with software engineering and development. There are three approaches to reliability.

1. Error avoidance:

Prevents errors from occurring in software

2. Error detection and correction:

In this approach errors are recognized whenever they are encountered and correcting the error by effect of error, of the system does not fail.

3. Error tolerance:

In this approach errors are recognized whenever they occur, but enable the system to keep running through degraded performance or by applying values that instruct the system to continue process.

Maintenance:

The key to reducing need for maintenance, while working, if possible to do essential tasks.

1. More accurately defining user requirement during system development.
2. Assembling better systems documentation.
3. Using more effective methods for designing, processing, login and communicating information with project team members.
4. Making better use of existing tools and techniques.
5. Managing system engineering process effectively.

Output Design:

One of the most important factors of an information system for the user is the output the system produces. Without the quality of the output, the entire system may appear unnecessary that will make us avoid using it possibly causing it to fail. Designing the output should process the in an organized well throughout the manner. The right output must be developed while ensuring that each output element is designed so that people will find the system easy to use effectively.

The term output applying to information produced by an information system whether printed or displayed while designing the output

we should identify the specific output that is needed to information requirements select a method to present the formation and create a document report or other formats that contains produced by the system.

Types of output:

Whether the output is formatted report or a simple listing of the contents of a file, a computer process will produce the output.

- A Document
- A Message
- Retrieval from a data store
- Transmission from a process or system activity
- Directly from an output sources

Layout Design:

It is an arrangement of items on the output medium. The layouts are building a mock up of the actual reports or document, as it will appear after the system is in operation. The output layout has been designated to cover information. The outputs are presented in the appendix.

Input design and control:

Input specifications describe the manner in which data enter the system for processing. Input design features will ensure the reliability of the systems and produce results from accurate data, or thus can be result in the production of erroneous information. The input design also determines whenever the user can interact efficiently with this system.

Objectives of input design:

Input design consists of developing specifications and procedures for data preparation, the steps necessary to put transaction data into a usable form for processing and data entry, the activity of data into the computer processing. The five objectives of input design are:

- Controlling the amount of input
- Avoiding delay
- Avoiding error in data
- Avoiding extra steps
- Keeping the process simple

Controlling the amount of input:

Data preparation and data entry operation depend on people, because labor costs are high, the cost of preparing and entering data is also high. Reducing data requirement expense. By reducing input requirement the speed of entire process from data capturing to processing to provide results to users.

Avoiding delay:

The processing delay resulting from data preparation or data entry operations is called bottlenecks. Avoiding bottlenecks should be one objective of input.

Avoiding errors:

Through input validation we control the errors in the input data.

Avoiding extra steps:

The designer should avoid the input design that cause extra steps in processing saving or adding a single step in large number of transactions saves a lot of processing time or takes more time to process.

Keeping process simple:

If controls are more people may feel difficult in using the systems. The best-designed system fits the people who use it in a way that is comfortable for them.

NORMALIZATION

It is a process of converting a relation to a standard form. The process is used to handle the problems that can arise due to data redundancy i.e. repetition of data in the database, maintain data integrity as well as handling problems that can arise due to insertion, updation, deletion anomalies.

Decomposing is the process of splitting relations into multiple relations to eliminate anomalies and maintain anomalies and maintain data integrity. To do this we use normal forms or rules for structuring relation.

Insertion anomaly:

Inability to add data to the database due to absence of other data.

Deletion anomaly:

Unintended loss of data due to deletion of other data.

Update anomaly:

Data inconsistency resulting from data redundancy and partial update

Normal Forms:

These are the rules for structuring relations that eliminate anomalies.

First Normal Form:

A relation is said to be in first normal form if the values in the relation are atomic for every attribute in the relation. By this we mean simply that no attribute value can be a set of values or, as it is sometimes expressed, a repeating group.

Second Normal Form:

A relation is said to be in second Normal form if it is in first normal form and it should satisfy any one of the following rules.

- 1) Primary key is not a composite primary key
- 2) No non key attributes are present
- 3) Every non key attribute is fully functionally dependent on full set of primary key.

Third Normal Form:

A relation is said to be in third normal form if there exists no transitive dependencies.

Transitive Dependency:

If two non key attributes depend on each other as well as on the primary key then they are said to be transitively dependent.

The above normalization principles were applied to decompose the data in multiple tables thereby making the data to be maintained in a consistent state.

MODULARITY

Software architecture embodies modularity; that is , software is divided into separately named and addressable components , often called modules ,that are integrated to satisfy problem requirements.

It has been stated that “modularity is the single attribute of software that allows a program to be intellectual manageable”. The number of control paths, span of reference, number of variables and over all complexity would make understand close to impossible.

Modular interaction

This application is for fulfilling the different requirements of the IT Staff and clients for an organization. With this application client should be in a position to do the following things:

1. Obtain General information about the Company.
2. Login into the Online Assistance System.
3. Create the new ticket.

4. Provide additional information about the ticket in the form of comments or attachments to the company.
5. Obtain details of the tickets in the form of active log provided by the helpdesk or administrator.

The main modules of this application are

- ❖ Ticket Submitters.
- ❖ Help Desk
- ❖ Administration.

Ticket Submitters

Ticket is a request, question or problem requested by the client to the company in the form of email, phone call, or by entering the information using the user interface. If the user wants to submit the ticket using the website, he / she must be registered with the site. Then only site allows submitting the ticket. If the ticket is submitted using the site, the project by defaultly assigns the TICKET SUBMITTERS role to the user. The advantage of this is user can view the active log, add new comments to the ticket and send any additional information to the request in the form of attachments.

Ticket Submitters can also have the privilege to provide the solution for the ticket and close the ticket. Other than this Ticket Submitters can search for the other tickets and view its status by clicking on the ticket link.

Ticket Submitters can have the following privileges:

1. Search for the Opened Tickets
2. View the status of any Tickets.
3. Add the comments to the tickets.
4. Close the ticket or re-open the tickets.

2. Help Desk

Help Desk is a group of experts (specialists) in different categories. These experts will provide the solutions to the tickets. Help Desk is having more privileges compare to the ticket submitters. Help Desk can handle any ticket to provide the solution and they can transfer the ticket to other Help Desk Expert. Help Desk can open the closed tickets, take over, reassign the tickets or they can close the ticket after providing the solution. Help Desk experts can directly communicate with clients or ticket submitters for any additional information required while providing the solution.

Specialists resolve those problems within their area of expertise and which they have been assigned. The escalation path identified for each problem report lists an assignee specialist (the assigned to field) and the subsequent specialists to whom the problem will be sent should it remain unresolved within a set time period.

If the problem can be solved immediately the specialist provides the immediate solution to the problem and closes the ticket with the problem resolution. If the problem cannot be solved immediately, the primary specialist create trouble ticket, record for the problem and he may contact the client or ticket submitter for more details to resolve the problem.

The specialist refers the checklist to resolve the problem along with the problem description he got from the call initiator. The problem description will be passed to the secondary specialist, in case the problem was not solved by the primary specialist. The secondary specialist uses the specialty escalation path to resolve the issue. He may also consider the ticket list and problem record for any further clarification; Based on the output the management in the support center resolves the problem. All these resolutions then passed to Administration department for analysis and after reviewing if necessary then will update it to the active log.

Administrator

Administrator is the main controller of the web site. He is having the privileges to control the users and their roles, site settings and if the problem is not solved by help desk specialists' admin is responsible for providing the solution for the ticket.

Administrator role In managing Users and web site.

1. By defaultly when a new user is registered with the web site. By defaultly website assigns "Ticket Submitter" role to the user. The default role can be changed to HELP DESK, ADMINISTRATOR or for one user both the roles can be assigned. So that user can act as a help desk expert and administrator.
2. Administrator can add or modify the ticket categories like (Hardware, software, technical, non technical, Network etc.
3. Ticket priorities like High, Low, Medium etc.
4. Ticket Types like Question, problem, request etc.

5. Administrator is responsible for assigning the tickets to experts.
6. Even admin also responsible to providing the solution for the tickets and the tickets which are not solved by the help desk experts.

DATA DICTIONARY

Data Dictionary

Table - dbo.Settings			
	Column Name	Data Type	Allow Nulls
▶	SettingName	nvarchar(50)	<input type="checkbox"/>
	SettingValue	nvarchar(MAX)	<input type="checkbox"/>

Table - dbo.TicketAttachments			
	Column Name	Data Type	Allow Nulls
▶	TicketId	int	<input type="checkbox"/>
▶	FileId	int	<input type="checkbox"/>
	FileName	nvarchar(255)	<input type="checkbox"/>
	FileSize	int	<input type="checkbox"/>
	FileType	nvarchar(50)	<input type="checkbox"/>
	UploadedBy	nvarchar(100)	<input type="checkbox"/>
	UploadDate	datetime	<input type="checkbox"/>
	FileContents	varbinary(MAX)	<input type="checkbox"/>

Table - dbo.TicketComments			
	Column Name	Data Type	Allow Nulls
▶?	TicketId	int	<input type="checkbox"/>
?	CommentId	int	<input type="checkbox"/>
	CommentEvent	nvarchar(500)	<input checked="" type="checkbox"/>
	Comment	ntext	<input checked="" type="checkbox"/>
	IsHtml	bit	<input type="checkbox"/>
	CommentedBy	nvarchar(100)	<input type="checkbox"/>
	CommentedDate	datetime	<input type="checkbox"/>
	Version	timestamp	<input type="checkbox"/>
			<input type="checkbox"/>

Table - dbo.Tickets			
	Column Name	Data Type	Allow Nulls
▶?	TicketId	int	<input type="checkbox"/>
	Type	nvarchar(50)	<input type="checkbox"/>
	Category	nvarchar(50)	<input type="checkbox"/>
	Title	nvarchar(500)	<input type="checkbox"/>
	Details	ntext	<input type="checkbox"/>
	IsHtml	bit	<input type="checkbox"/>
	TagList	nvarchar(100)	<input checked="" type="checkbox"/>
	CreatedBy	nvarchar(100)	<input type="checkbox"/>
	CreatedDate	datetime	<input type="checkbox"/>
	Owner	nvarchar(100)	<input type="checkbox"/>
	AssignedTo	nvarchar(100)	<input checked="" type="checkbox"/>
	CurrentStatus	nvarchar(50)	<input type="checkbox"/>
	CurrentStatusDate	datetime	<input type="checkbox"/>
	CurrentStatusSetBy	nvarchar(100)	<input type="checkbox"/>
	LastUpdateBy	nvarchar(100)	<input type="checkbox"/>
	LastUpdateDate	datetime	<input type="checkbox"/>
	Priority	nvarchar(25)	<input checked="" type="checkbox"/>
	AffectsCustomer	bit	<input type="checkbox"/>
	PublishedToKb	bit	<input type="checkbox"/>
	Version	timestamp	<input type="checkbox"/>
			<input type="checkbox"/>

Table - dbo.TicketTags			
	Column Name	Data Type	Allow Nulls
▶?	TicketTagId	int	<input type="checkbox"/>
	TagName	nvarchar(100)	<input type="checkbox"/>
	TicketId	int	<input type="checkbox"/>
			<input type="checkbox"/>

Table - dbo.aspnet_Applications			
	Column Name	Data Type	Allow Nulls
▶	ApplicationName	nvarchar(256)	<input type="checkbox"/>
	LoweredApplicationName	nvarchar(256)	<input type="checkbox"/>
?	ApplicationId	uniqueidentifier	<input type="checkbox"/>
	Description	nvarchar(256)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Table - dbo.aspnet_Membership			
	Column Name	Data Type	Allow Nulls
?	UserId	uniqueidentifier	<input type="checkbox"/>
	Password	nvarchar(128)	<input type="checkbox"/>
	PasswordFormat	int	<input type="checkbox"/>
	PasswordSalt	nvarchar(128)	<input type="checkbox"/>
	MobilePIN	nvarchar(16)	<input checked="" type="checkbox"/>
	Email	nvarchar(256)	<input checked="" type="checkbox"/>
	LoweredEmail	nvarchar(256)	<input checked="" type="checkbox"/>
	PasswordQuestion	nvarchar(256)	<input checked="" type="checkbox"/>
	PasswordAnswer	nvarchar(128)	<input checked="" type="checkbox"/>
	IsApproved	bit	<input type="checkbox"/>
	IsLockedOut	bit	<input type="checkbox"/>
	CreateDate	datetime	<input type="checkbox"/>
	LastLoginDate	datetime	<input type="checkbox"/>
	LastPasswordChangedDate	datetime	<input type="checkbox"/>
	LastLockoutDate	datetime	<input type="checkbox"/>
	FailedPasswordAttemptCount	int	<input type="checkbox"/>
	FailedPasswordAttemptWin...	datetime	<input type="checkbox"/>
	FailedPasswordAnswerAtte...	int	<input type="checkbox"/>
	FailedPasswordAnswerAtte...	datetime	<input type="checkbox"/>
	Comment	ntext	<input checked="" type="checkbox"/>

Table - dbo.aspnet_Paths			
	Column Name	Data Type	Allow Nulls
▶	ApplicationId	uniqueidentifier	<input type="checkbox"/>
?	PathId	uniqueidentifier	<input type="checkbox"/>
	Path	nvarchar(256)	<input type="checkbox"/>
	LoweredPath	nvarchar(256)	<input type="checkbox"/>
			<input type="checkbox"/>

Table - dbo.aspne...alizationAllUsers			
	Column Name	Data Type	Allow Nulls
▶🔑	PathId	uniqueidentifier	<input type="checkbox"/>
	PageSettings	image	<input type="checkbox"/>
	LastUpdatedDate	datetime	<input type="checkbox"/>

Table - dbo.aspne...alizationAllUsers			
	Column Name	Data Type	Allow Nulls
▶🔑	PathId	uniqueidentifier	<input type="checkbox"/>
	PageSettings	image	<input type="checkbox"/>
	LastUpdatedDate	datetime	<input type="checkbox"/>

Table - dbo.aspnet_Profile			
	Column Name	Data Type	Allow Nulls
▶🔑	UserId	uniqueidentifier	<input type="checkbox"/>
	PropertyNames	ntext	<input type="checkbox"/>
	PropertyValuesString	ntext	<input type="checkbox"/>
	PropertyValuesBinary	image	<input type="checkbox"/>
	LastUpdatedDate	datetime	<input type="checkbox"/>

Table - dbo.aspnet_Roles			
	Column Name	Data Type	Allow Nulls
▶	ApplicationId	uniqueidentifier	<input type="checkbox"/>
🔑	RoleId	uniqueidentifier	<input type="checkbox"/>
	RoleName	nvarchar(256)	<input type="checkbox"/>
	LoweredRoleName	nvarchar(256)	<input type="checkbox"/>
	Description	nvarchar(256)	<input checked="" type="checkbox"/>

Table - dbo.asp...t_SchemaVersions			
	Column Name	Data Type	Allow Nulls
▶🔑	Feature	nvarchar(128)	<input type="checkbox"/>
🔑	CompatibleSchemaVersion	nvarchar(128)	<input type="checkbox"/>
	IsCurrentVersion	bit	<input type="checkbox"/>

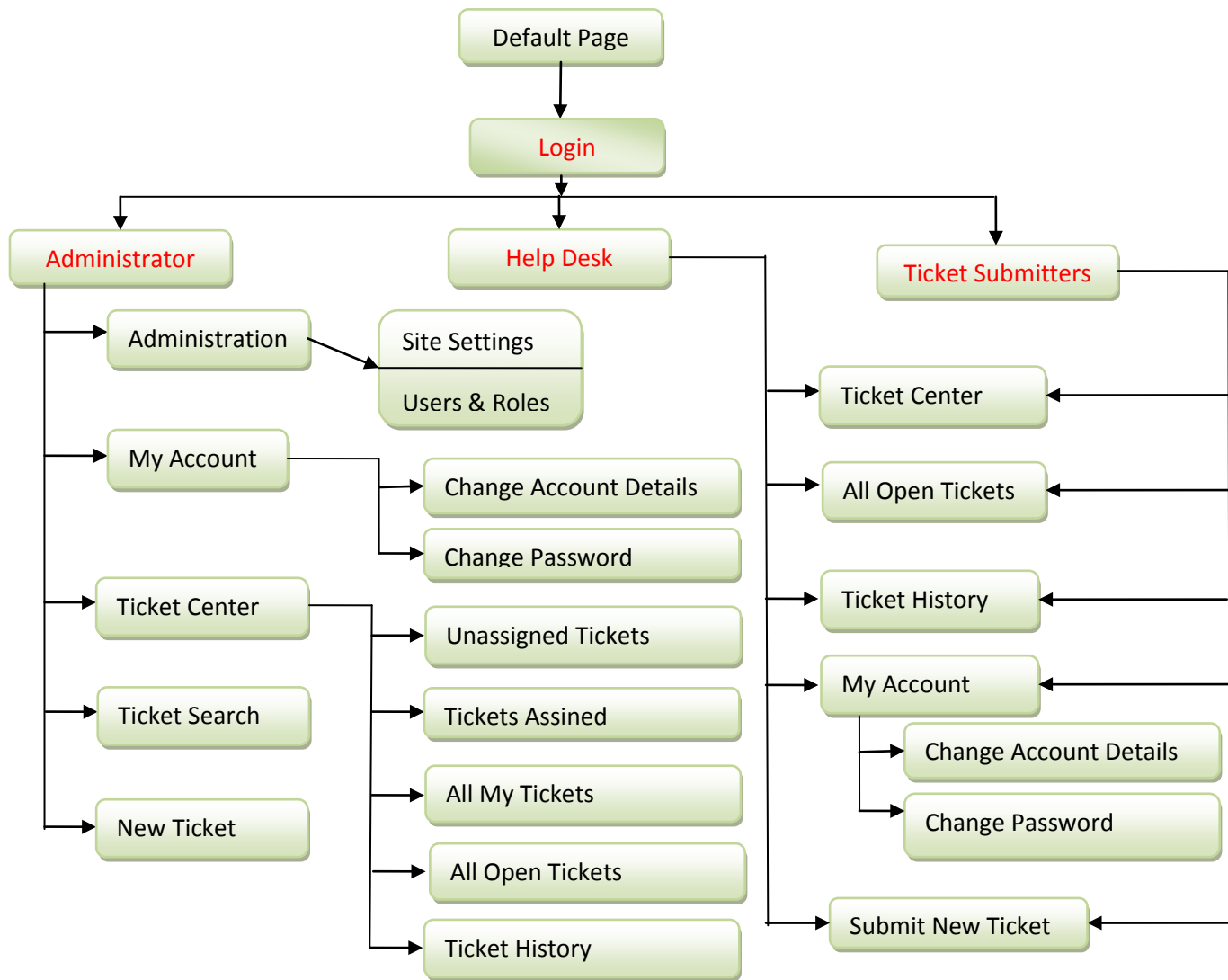
Table - dbo.aspnet_Users			
	Column Name	Data Type	Allow Nulls
►	ApplicationId	uniqueidentifier	<input type="checkbox"/>
🔑	UserId	uniqueidentifier	<input type="checkbox"/>
	UserName	nvarchar(256)	<input type="checkbox"/>
	LoweredUserName	nvarchar(256)	<input type="checkbox"/>
	MobileAlias	nvarchar(16)	<input checked="" type="checkbox"/>
	IsAnonymous	bit	<input type="checkbox"/>
	LastActivityDate	datetime	<input type="checkbox"/>

Table - dbo.aspnet_UsersInRoles			
	Column Name	Data Type	Allow Nulls
►🔑	UserId	uniqueidentifier	<input type="checkbox"/>
🔑	RoleId	uniqueidentifier	<input type="checkbox"/>

Table - dbo.aspnet_WebEvent_Events			
	Column Name	Data Type	Allow Nulls
►🔑	EventId	char(32)	<input type="checkbox"/>
	EventTimeUtc	datetime	<input type="checkbox"/>
	EventTime	datetime	<input type="checkbox"/>
	EventType	nvarchar(256)	<input type="checkbox"/>
	EventSequence	decimal(19, 0)	<input type="checkbox"/>
	EventOccurrence	decimal(19, 0)	<input type="checkbox"/>
	EventCode	int	<input type="checkbox"/>
	EventDetailCode	int	<input type="checkbox"/>
	Message	nvarchar(1024)	<input checked="" type="checkbox"/>
	ApplicationPath	nvarchar(256)	<input checked="" type="checkbox"/>
	ApplicationVirtualPath	nvarchar(256)	<input checked="" type="checkbox"/>
	MachineName	nvarchar(256)	<input type="checkbox"/>
	RequestUrl	nvarchar(1024)	<input checked="" type="checkbox"/>
	ExceptionType	nvarchar(256)	<input checked="" type="checkbox"/>
	Details	ntext	<input checked="" type="checkbox"/>

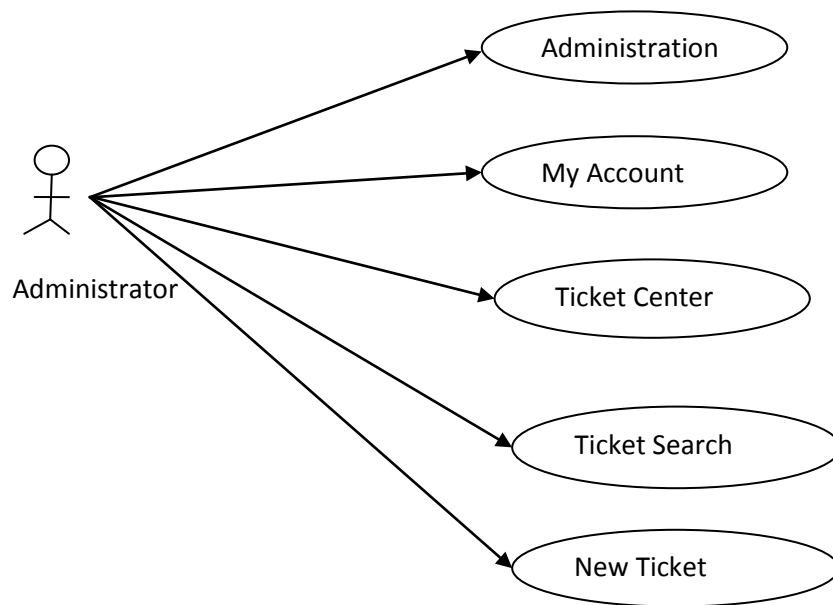
UML DIAGRAMS

Context Level Diagram

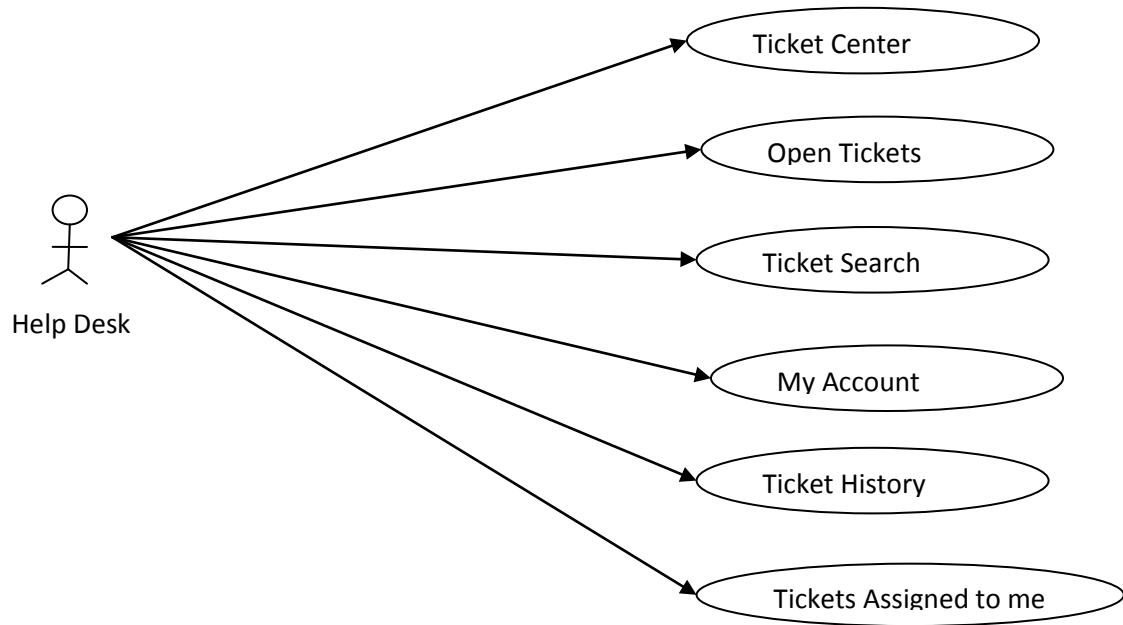


Use Case Diagrams

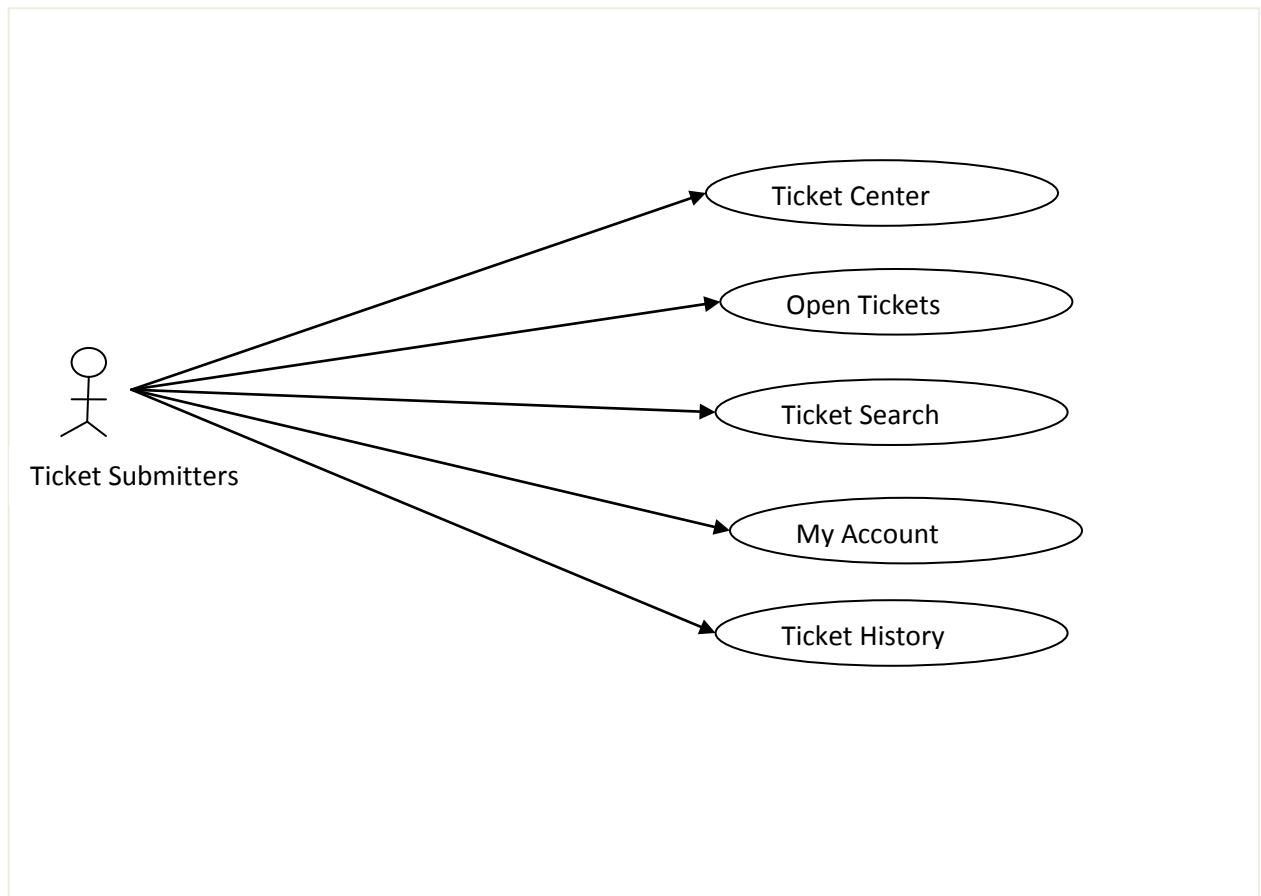
Administrator Use case



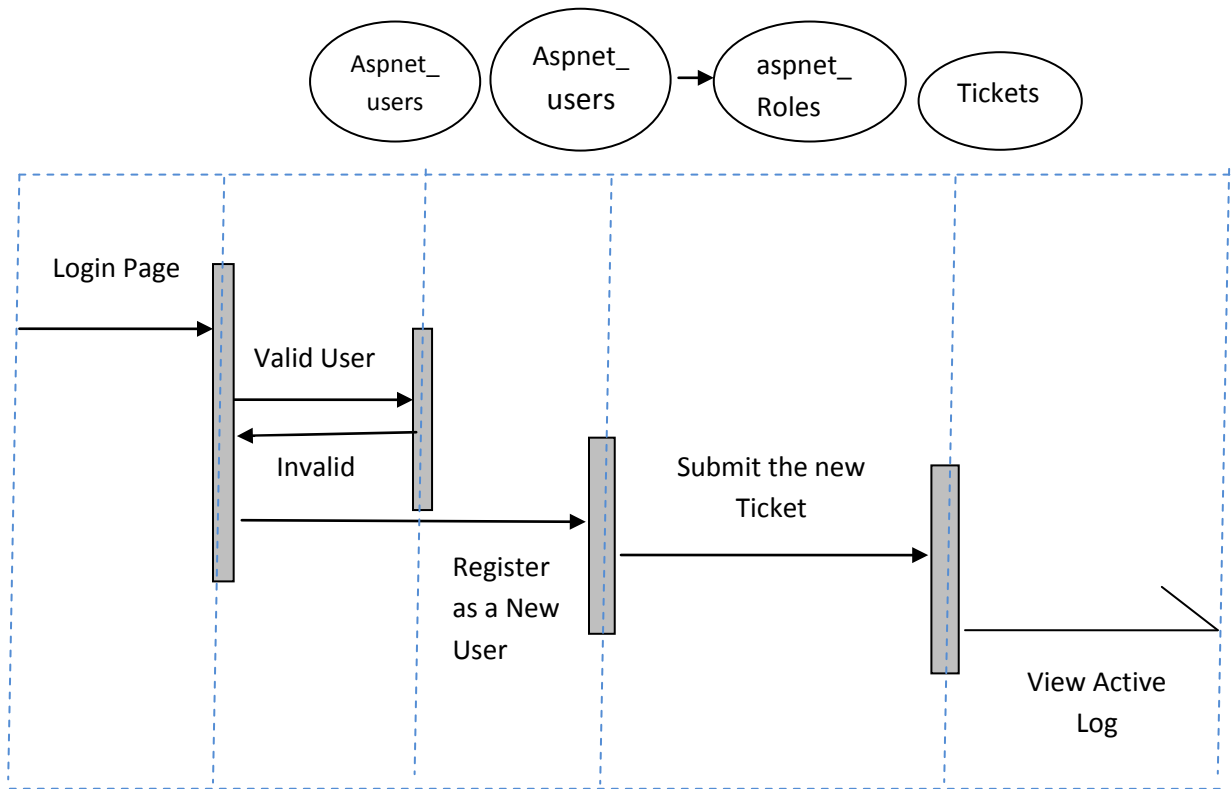
Help Desk Use case Diagram



Use Case Diagram for Ticket Submitters

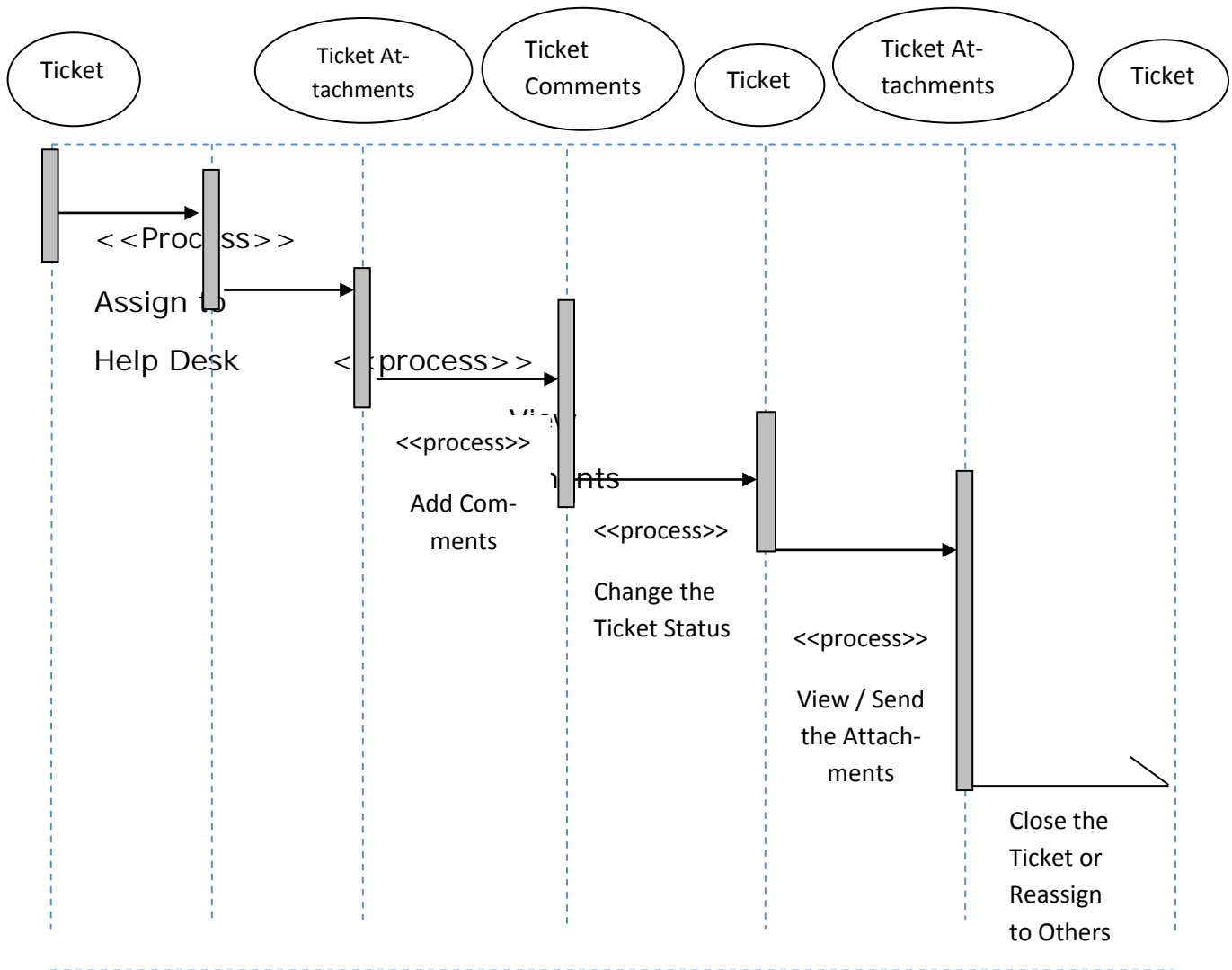


Sequence Diagram for Login Page.



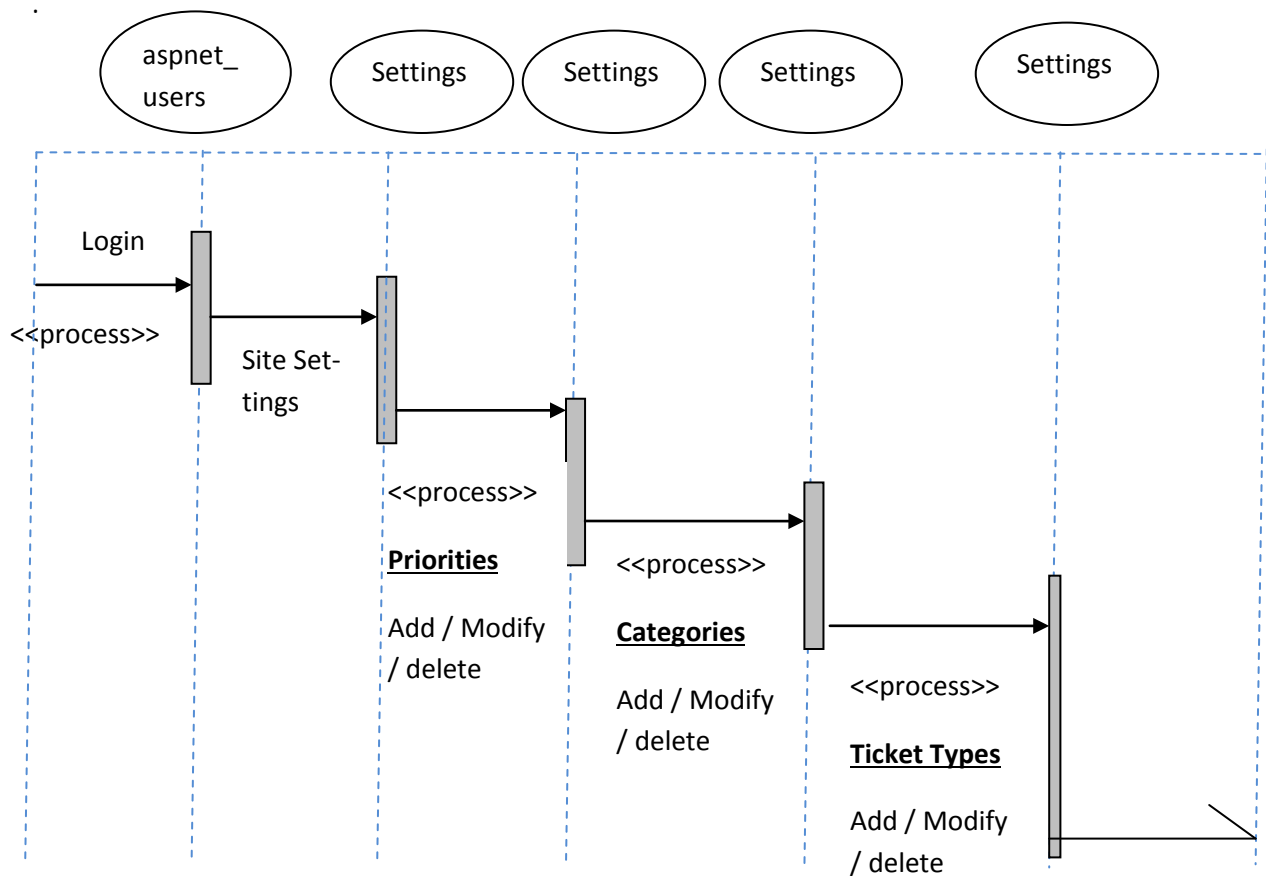
By default whenever new user is registered, System assigns the TICKET SUBMITTER Role to the user. Later Administrator can change the role of the user as Administrator, Help Desk or Ticket Submitter.

Sequence Diagram for Ticket.

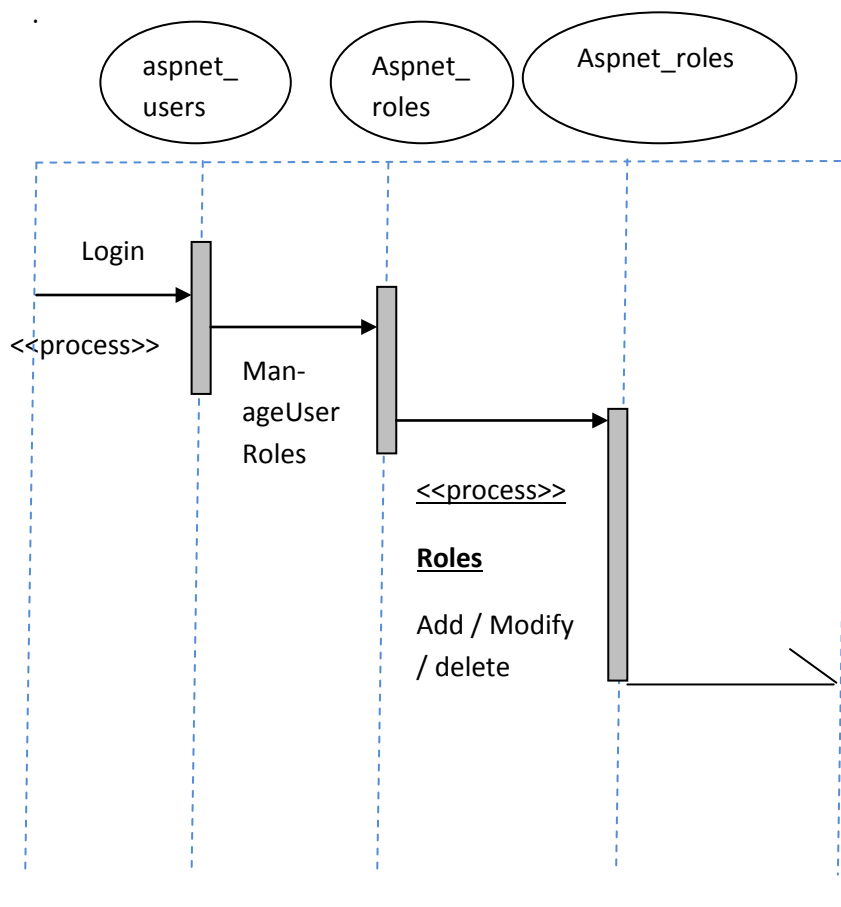


The process of assigning the ticket to others can be done by Administrator, Help Desk and Ticket Submitters. But closing the ticket can be done by Helpdesk or Administrator Only.

Administrator and site setting Sequence diagram.

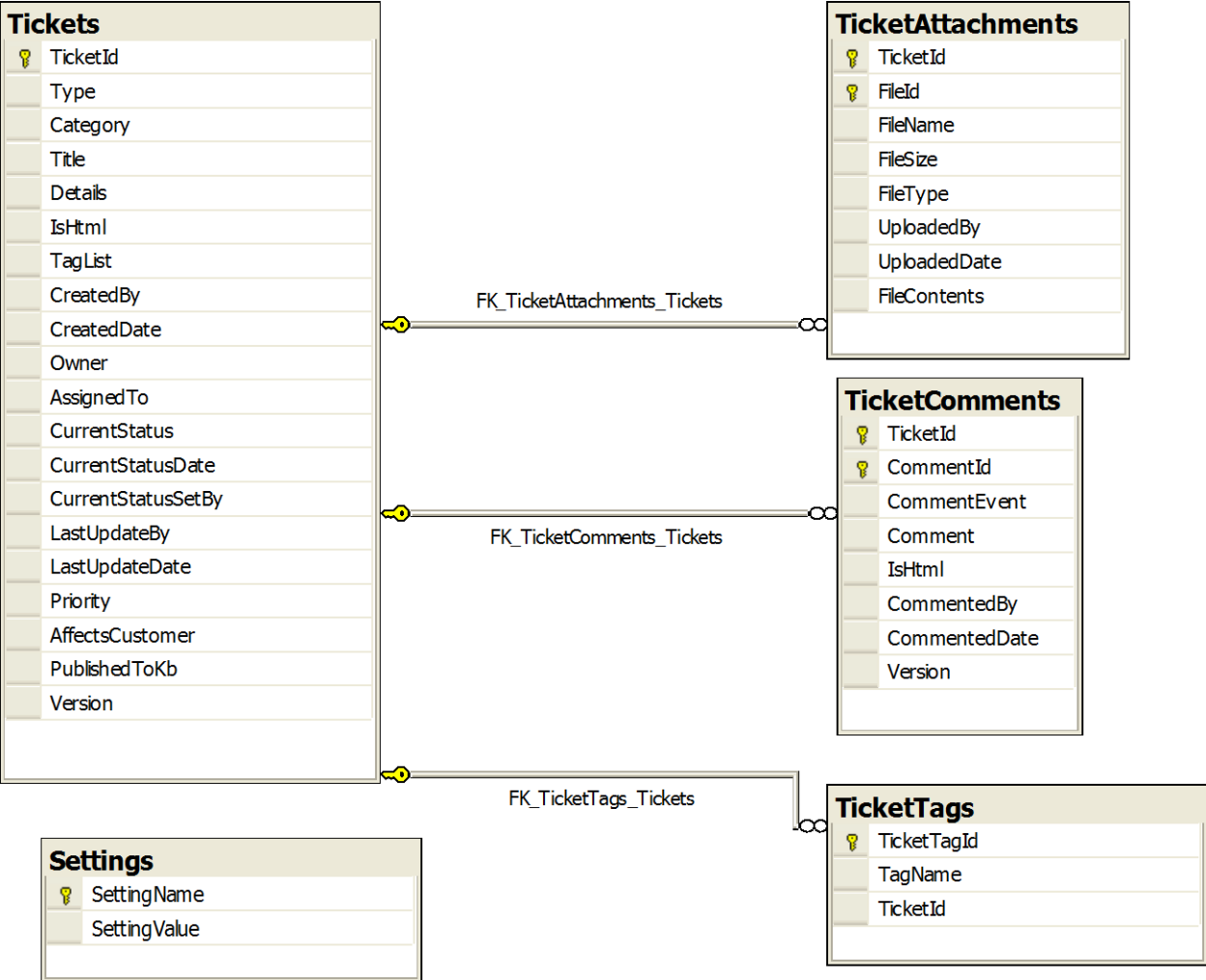


Administrator - Users & Roles Sequence Diagram.



E-R DIAGRAM

ER Diagram



8. SYSTEM DEVELOPMENT

CONCEPTS AND TECHNIQUES

Microsoft.NET Framework

The .NET Framework is a new computing platform that simplifies application development in the highly distributed environment of the Internet. The .NET Framework is designed to fulfill the following objectives:

- To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but Internet-distributed, or executed remotely.
- To provide a code-execution environment that minimizes software deployment and versioning conflicts.
- To provide a code-execution environment that guarantees safe execution of code, including code created by an unknown or semi-trusted third party.
- To provide a code-execution environment that eliminates the performance problems of scripted or interpreted environments.

- To make the developer experience consistent across widely varying types of applications, such as Windows-based applications and Web-based applications.
- To build all communication on industry standards to ensure that code based on the .NET Framework can integrate with any other code.

The .NET Framework has two main components: the common language runtime and the .NET Framework class library. The common language runtime is the foundation of the .NET Framework. You can think of the runtime as an agent that manages code at execution time, providing core services such as memory management, thread management, and remoting, while also enforcing strict type safety and other forms of code accuracy that ensure security and robustness. In fact, the concept of code management is a fundamental principle of the runtime. Code that targets the runtime is known as managed code, while code that does not target the runtime is known as unmanaged code. The class library, the other main component of the .NET Framework, is a comprehensive, object-oriented collection of reusable types that you can use to develop applications ranging from traditional command-line or graphical user interface (GUI) applications to applications based on the latest innovations provided by ASP.NET, such as Web Forms and XML Web services.

The .NET Framework can be hosted by unmanaged components that load the common language runtime into their processes and initiate the execution of managed code, thereby creating a software environment that can exploit both managed and unmanaged features. The .NET Framework not only provides several runtime hosts, but also supports the development of third-party runtime hosts.

For example, ASP.NET hosts the runtime to provide a scalable, server-side environment for managed code. ASP.NET works directly with the runtime to enable Web Forms applications and XML Web services, both of which are discussed later in this topic.

Internet Explorer is an example of an unmanaged application that hosts the runtime (in the form of a MIME type extension). Using Internet Explorer to host the runtime enables you to embed managed components or Web Forms controls in HTML documents. Hosting the runtime in this way makes managed mobile code (similar to Microsoft® ActiveX® controls) possible, but with significant improvements that only managed code can offer, such as semi-trusted execution and secure isolated file storage.

The following illustration shows the relationship of the common language runtime and the class library to your applications and to the

overall system. The illustration also shows how managed code operates within a larger architecture.

Features of the Common Language Runtime

The common language runtime manages memory, thread execution, code execution, code safety verification, compilation, and other system services. These features are intrinsic to the managed code that runs on the common language runtime.

With regards to security, managed components are awarded varying degrees of trust, depending on a number of factors that include their origin (such as the Internet, enterprise network, or local computer). This means that a managed component might or might not be able to perform file-access operations, registry-access operations, or other sensitive functions, even if it is being used in the same active application.

The runtime enforces code access security. For example, users can trust that an executable embedded in a Web page can play an animation on screen or sing a song, but cannot access their personal data, file system, or network. The security features of the runtime thus enable legitimate Internet-deployed software to be exceptionally feature rich.

The runtime also enforces code robustness by implementing a strict type- and code-verification infrastructure called the common type system (CTS). The CTS ensures that all managed code is self-describing. The various Microsoft and third-party language compilers generate managed code that conforms to the CTS. This means that managed code can consume other managed types and instances, while strictly enforcing type fidelity and type safety.

In addition, the managed environment of the runtime eliminates many common software issues. For example, the runtime automatically handles object layout and manages references to objects, releasing them when they are no longer being used. This automatic memory management resolves the two most common application errors, memory leaks and invalid memory references.

The runtime also accelerates developer productivity. For example, programmers can write applications in their development language of choice, yet take full advantage of the runtime, the class library, and components written in other languages by other developers. Any compiler vendor who chooses to target the runtime can do so. Language compilers that target the .NET Framework make the features of the .NET Framework available to existing code written in that language, greatly easing the migration process for existing applications.

While the runtime is designed for the software of the future, it also supports software of today and yesterday. Interoperability between managed and unmanaged code enables developers to continue to use necessary COM components and DLLs.

The runtime is designed to enhance performance. Although the common language runtime provides many standard runtime services, managed code is never interpreted. A feature called just-in-time (JIT) compiling enables all managed code to run in the native machine language of the system on which it is executing. Meanwhile, the memory manager removes the possibilities of fragmented memory and increases memory locality-of-reference to further increase performance.

Finally, the runtime can be hosted by high-performance, server-side applications, such as Microsoft® SQL Server™ and Internet Information Services (IIS). This infrastructure enables you to use managed code to write your business logic, while still enjoying the superior performance of the industry's best enterprise servers that support runtime hosting.

.NET Framework Class Library

The .NET Framework class library is a collection of reusable types that tightly integrate with the common language runtime. The class library is object oriented, providing types from which your own managed code

can derive functionality. This not only makes the .NET Framework types easy to use, but also reduces the time associated with learning new

features of the .NET Framework. In addition, third-party components can integrate seamlessly with classes in the .NET Framework.

For example, the .NET Framework collection classes implement a set of interfaces that you can use to develop your own collection classes. Your collection classes will blend seamlessly with the classes in the .NET Framework.

As you would expect from an object-oriented class library, the .NET Framework types enable you to accomplish a range of common programming tasks, including tasks such as string management, data collection, database connectivity, and file access. In addition to these common tasks, the class library includes types that support a variety of specialized development scenarios. For example, you can use the .NET Framework to develop the following types of applications and services:

- Console applications.
- Scripted or hosted applications.
- Windows GUI applications (Windows Forms).
- ASP.NET applications.

- XML Web services.
- Windows services.

For example, the Web Forms classes are a comprehensive set of reusable types that vastly simplify Web GUI development. If you write an ASP.NET Web Form application, you can use the Web Forms classes.

C#.NET

Introduction to Web Forms (C#.NET)

Web Forms is the new platform for Microsoft Web application development, based on the .NET Framework. This framework provides a clear, object-oriented, extensible set of classes that enable you to develop rich Web applications. Additionally, Web Forms can act as the local user interface in a multi-tier distributed solution. Web Forms is a framework for building Windows client applications that utilize the common language runtime. Web Forms applications can be written in any language that the common language runtime supports.

What Is a Web Form?

Web Forms have the potential to change Web programming by introducing a new programming model built around server-side controls—a

model in which controls render their own UIs by generating HTML to return to clients and firing events that are handled by server-side scripts. Since all the action takes place on the Web server, virtually any browser can run a Web Forms app. And thanks to Visual Studio.NET, building a Web Forms app is a lot like using Visual Basic: just drop a control onto a form then write an event handler. This article describes the Web Forms programming model, how it fits with ASP.NET, and introduces DataGrid, TextBox, and other classes in the .NET Framework class library.

As with all objects in the .NET Framework, forms are instances of classes. The form you create with the web Forms Designer is a class, and when you display an instance of the form at run time, this class is the template used to create the form. The framework also allows you to inherit from existing forms to add functionality or modify existing behavior. When you add a form to your project, you can choose whether it inherits from the **Form** class provided by the framework, or from a form you have previously created.

Additionally, forms are controls, because they inherit from the Control class.

Within a Web Forms project, the form is the primary vehicle for user interaction. By combining different sets of controls and writing code,

you can elicit information from the user and respond to it, work with existing stores of data, and query and write back to the file system and registry on the user's local computer.

Although the form can be created entirely in the Code Editor, it is easier to use the Web Forms Designer to create and modify forms.

Some of the advantages of using Web Forms include the following:

- **Simplicity and power:** Web Forms is a programming model for developing Web applications that combines the simplicity of the Visual Basic programming model with the power and flexibility of the common language runtime.
- **Lower total cost of ownership:** Web Forms takes advantage of the versioning and deployment features of the common language runtime to offer reduced deployment costs and higher application robustness over time. This significantly lowers the maintenance costs (TCO) for applications written in Web Forms.
- **Architecture for controls:** Web Forms offers an architecture for controls and control containers that is based on concrete implementation of the control and container classes. This significantly reduces control-container interoperability issues.
- **Security:** Web Forms takes full advantage of the security features of the common language runtime. This means that Web Forms can be used implement everything from an untrusted control running in the browser to a fully trusted application installed on a user's hard disk.

- **XML Web services support:** Web Forms offers full support for quickly and easily connecting to XML Web services.
- **Rich graphics:** Web Forms is one of the first ship vehicles for GDI+, a new version of the Web Graphical Device Interface (GDI) that supports alpha blending, texture brushes, advanced transforms, rich text support, and more.
- **Flexible controls:** Web Forms offers a rich set of controls that encompass all of the controls offered by Web. These controls also offer new features, such as "flat look" styles for buttons, radio buttons, and check boxes.
- **Data awareness:** Web Forms offers full support for the ADO data model.
- **ActiveX control support:** Web Forms offers full support for ActiveX controls. You can easily host ActiveX controls in a Web Forms application. You can also host a Web Forms control as an ActiveX control.
- **Licensing:** Web Forms takes advantage of the common language runtime enhanced licensing model.
- **Printing:** Web Forms offers a printing framework that enables applications to provide comprehensive reports.
- **Accessibility:** Web Forms controls implement the interfaces defined by Microsoft Active Accessibility (MSAA), which make it simple to build applications that support accessibility aids, such as screen readers.
- **Design-time support:** Web Forms takes full advantage of the meta-data and component model features offered by the common language runtime to provide thorough design-time support for both control users and control implementers.

ACTIVE X DATA OBJECTS.NET

ADO.NET Overview

ADO.NET is an evolution of the ADO data access model that directly addresses user requirements for developing scalable applications. It was designed specifically for the web with scalability, statelessness, and XML in mind.

ADO.NET uses some ADO objects, such as the **Connection** and **Command** objects, and also introduces new objects. Key new ADO.NET objects include the **DataSet**, **DataReader**, and **DataAdapter**.

The important distinction between this evolved stage of ADO.NET and previous data architectures is that there exists an object -- the **DataSet** -- that is separate and distinct from any data stores. Because of that, the **DataSet** functions as a standalone entity. You can think of the **DataSet** as an always disconnected recordset that knows nothing about the source or destination of the data it contains. Inside a **DataSet**, much like in a database, there are tables, columns, relationships, constraints, views, and so forth.

A **DataAdapter** is the object that connects to the database to fill the **DataSet**. Then, it connects back to the database to update the data there, based on operations performed while the **DataSet** held the data. In the past, data processing has been primarily connection-based. Now, in an effort to make multi-tiered apps more efficient, data processing is turning to a message-based approach that revolves around

chunks of information. At the center of this approach is the **DataAdapter**, which provides a bridge to retrieve and save data between a **DataSet** and its source data store. It accomplishes this by means of requests to the appropriate SQL commands made against the data store.

The XML-based **DataSet** object provides a consistent programming model that works with all models of data storage: flat, relational, and hierarchical. It does this by having no 'knowledge' of the source of its data, and by representing the data that it holds as collections and data types. No matter what the source of the data within the **DataSet** is, it is manipulated through the same set of standard APIs exposed through the **DataSet** and its subordinate objects.

While the **DataSet** has no knowledge of the source of its data, the managed provider has detailed and specific information. The role of the managed provider is to connect, fill, and persist the **DataSet** to and from data stores. The OLE DB and MSACCESS .NET Data Providers (System.Data.OleDb and System.Data.SqlClient) that are part of the .Net Framework provide four basic objects: the **Command**, **Connection**, **DataReader** and **DataAdapter**. In the remaining sections of this document, we'll walk through each part of the **DataSet** and the OLE DB/MSACCESS .NET Data Providers explaining what they are, and how to program against them.

The following sections will introduce you to some objects that have evolved, and some that are new. These objects are:

- **Connections**. For connection to and managing transactions against a database.
- **Commands**. For issuing SQL commands against a database.

- **DataReaders**. For reading a forward-only stream of data records from a MSACCESS data source.
- **DataSets**. For storing, remoting and programming against flat data, XML data and relational data.
- **DataAdapters**. For pushing data into a **DataSet**, and reconciling data against a database.

When dealing with connections to a database, there are two different options: MSACCESS .NET Data Provider (System.Data.SqlClient) and OLE DB .NET Data Provider (System.Data.OleDb). In these samples we will use the MSACCESS .NET Data Provider. These are written to talk directly to Microsoft MSACCESS. The OLE DB .NET Data Provider is used to talk to any OLE DB provider (as it uses OLE DB underneath).

Connections

Connections are used to 'talk to' databases, and are represented by provider-specific classes such as **SqlConnection**. Commands travel over connections and resultsets are returned in the form of streams which can be read by a **DataReader** object, or pushed into a **DataSet** object.

Commands

Commands contain the information that is submitted to a database, and are represented by provider-specific classes such as **SqlCommand**. A command can be a stored procedure call, an UPDATE statement, or a statement that returns results. You can also use input and output parameters, and return values as part of your command syntax. The example below shows how to issue an INSERT statement against the **Northwind** database.

DataReaders

The **DataReader** object is somewhat synonymous with a read-only/forward-only cursor over data. The **DataReader** API supports flat as well as hierarchical data. A **DataReader** object is returned after executing a command against a database. The format of the returned **DataReader** object is different from a recordset. For example, you might use the **DataReader** to show the results of a search list in a web page.

DataSets and DataAdapters

DataSets

The **DataSet** object is similar to the ADO **Recordset** object, but more powerful, and with one other important distinction: the **DataSet** is always disconnected. The **DataSet** object represents a cache of data, with database-like structures such as tables, columns, relationships, and constraints. However, though a **DataSet** can and does behave much like a database, it is important to remember that **DataSet** objects do not interact directly with databases, or other source data. This allows the developer to work with a programming model that is always consistent, regardless of where the source data resides. Data coming from a database, an XML file, from code, or user input can all be placed into **DataSet** objects. Then, as changes are made to the **DataSet** they can be tracked and verified before updating the source data. The **GetChanges** method of the **DataSet** object actually creates a second **DataSet** that contains only the changes to the data. This **DataSet** is then used by a **DataAdapter** (or other objects) to update the original data source.

The **DataSet** has many XML characteristics, including the ability to produce and consume XML data and XML schemas. XML schemas can be used to describe schemas interchanged via WebServices. In fact, a **DataSet** with a schema can actually be compiled for type safety and statement completion.

DataAdapters (OLEDB/SQL)

The **DataAdapter** object works as a bridge between the **DataSet** and the source data. Using the provider-specific **SqlDataAdapter** (along with its associated **SqlCommand** and **SqlConnection**) can increase overall performance when working with a Microsoft MSACCESS databases. For other OLE DB-supported databases, you would use the **OleDbDataAdapter** object and its associated **OleDbCommand** and **OleDbConnection** objects.

The **DataAdapter** object uses commands to update the data source after changes have been made to the **DataSet**. Using the **Fill** method of the **DataAdapter** calls the SELECT command; using the **Update** method calls the INSERT, UPDATE or DELETE command for each changed row. You can explicitly set these commands in order to control the statements used at runtime to resolve changes, including the use of stored procedures. For ad-hoc scenarios, a **CommandBuilder** object can generate these at run-time based upon a select statement. However, this run-time generation requires an extra round-trip to the server in order to gather required metadata, so explicitly providing the INSERT, UPDATE, and DELETE commands at design time will result in better run-time performance.

1. ADO.NET is the next evolution of ADO for the .Net Framework.

2. ADO.NET was created with n-Tier, statelessness and XML in the forefront. Two new objects, the **DataSet** and **DataAdapter**, are provided for these scenarios.
3. ADO.NET can be used to get data from a stream, or to store data in a cache for updates.
4. There is a lot more information about ADO.NET in the documentation.
5. Remember, you can execute a command directly against the database in order to do inserts, updates, and deletes. You don't need to first put data into a **DataSet** in order to insert, update, or delete it.
6. Also, you can use a **DataSet** to bind to the data, move through the data, and navigate data relationships

ASP.Net

Server Application Development

Server-side applications in the managed world are implemented through runtime hosts. Unmanaged applications host the common language runtime, which allows your custom managed code to control the behavior of the server. This model provides you with all the features of the common language runtime and class library while gaining the performance and scalability of the host server.

The following illustration shows a basic network schema with managed code running in different server environments. Servers such as IIS and MSACCESS can perform standard operations while your application logic executes through the managed code.

Server-side managed code

ASP.NET is the hosting environment that enables developers to use the .NET Framework to target Web-based applications. However, ASP.NET is more than just a runtime host; it is a complete architecture for developing Web sites and Internet-distributed objects using managed code. Both Web Forms and XML Web services use IIS and ASP.NET as the publishing mechanism for applications, and both have a collection of supporting classes in the .NET Framework.

XML Web services, an important evolution in Web-based technology, are distributed, server-side application components similar to common Web sites. However, unlike Web-based applications, XML Web services components have no UI and are not targeted for browsers such as Internet Explorer and Netscape Navigator. Instead, XML Web services consist of reusable software components designed to be consumed by other applications, such as traditional client applications, Web-based applications, or even other XML Web services. As a result, XML Web services technology is rapidly moving application development and deployment into the highly distributed environment of the Internet.

If you have used earlier versions of ASP technology, you will immediately notice the improvements that ASP.NET and Web Forms offers. For example, you can develop Web Forms pages in any language that supports the .NET Framework. In addition, your code no longer needs to share the same file with your HTTP text (although it can continue to

do so if you prefer). Web Forms pages execute in native machine language because, like any other managed application, they take full advantage of the runtime. In contrast, unmanaged ASP pages are always scripted and interpreted. ASP.NET pages are faster, more functional, and easier to develop than unmanaged ASP pages because they interact with the runtime like any managed application.

The .NET Framework also provides a collection of classes and tools to aid in development and consumption of XML Web services applications. XML Web services are built on standards such as SOAP (a remote procedure-call protocol), XML (an extensible data format), and WSDL (the Web Services Description Language). The .NET Framework is built on these standards to promote interoperability with non-Microsoft solutions.

For example, the Web Services Description Language tool included with the .NET Framework SDK can query an XML Web service published on the Web, parse its WSDL description, and produce C# or Visual Basic source code that your application can use to become a client of the XML Web service. The source code can create classes derived from classes in the class library that handle all the underlying communication using SOAP and XML parsing. Although you can use the class library to consume XML Web services directly, the Web Services Description Language tool and the other tools contained in the SDK facilitate your development efforts with the .NET Framework.

If you develop and publish your own XML Web service, the .NET Framework provides a set of classes that conform to all the underlying communication standards, such as SOAP, WSDL, and XML. Using those classes enables you to focus on the logic of your service, without con-

cerning yourself with the communications infrastructure required by distributed software development.

Finally, like Web Forms pages in the managed environment, your XML Web service will run with the speed of native machine language using the scalable communication of IIS.

Active Server Pages.NET

ASP.NET is a programming framework built on the common language runtime that can be used on a server to build powerful Web applications. ASP.NET offers several important advantages over previous Web development models:

- **Enhanced Performance.** ASP.NET is compiled common language runtime code running on the server. Unlike its interpreted predecessors, ASP.NET can take advantage of early binding, just-in-time compilation, native optimization, and caching services right out of the box. This amounts to dramatically better performance before you ever write a line of code.
- **World-Class Tool Support.** The ASP.NET framework is complemented by a rich toolbox and designer in the Visual Studio integrated development environment. WYSIWYG editing, drag-and-drop server controls, and automatic deployment are just a few of the features this powerful tool provides.
- **Power and Flexibility.** Because ASP.NET is based on the common language runtime, the power and flexibility of that entire

platform is available to Web application developers. The .NET Framework class library, Messaging, and Data Access solutions are all seamlessly accessible from the Web. ASP.NET is also language-independent, so you can choose the language that best applies to your application or partition your application across many languages. Further, common language runtime interoperability guarantees that your existing investment in COM-based development is preserved when migrating to ASP.NET.

- **Simplicity.** ASP.NET makes it easy to perform common tasks, from simple form submission and client authentication to deployment and site configuration. For example, the ASP.NET page framework allows you to build user interfaces that cleanly separate application logic from presentation code and to handle events in a simple, Visual Basic - like forms processing model. Additionally, the common language runtime simplifies development, with managed code services such as automatic reference counting and garbage collection.

- **Manageability.** ASP.NET employs a text-based, hierarchical configuration system, which simplifies applying settings to your server environment and Web applications. Because configuration information is stored as plain text, new settings may be applied without the aid of local administration tools. This "zero local administration" philosophy extends to deploying ASP.NET Framework applications as well. An ASP.NET Framework application is deployed to a server simply by copying the necessary files to the server. No server restart is required, even to deploy or replace running compiled code.

- **Scalability and Availability.** ASP.NET has been designed with scalability in mind, with features specifically tailored to improve performance in clustered and multiprocessor environments. Further,

processes are closely monitored and managed by the ASP.NET runtime, so that if one misbehaves (leaks, deadlocks), a new process can be created in its place, which helps keep your application constantly available to handle requests.

- **Customizability and Extensibility.** ASP.NET delivers a well-factored architecture that allows developers to "plug-in" their code at the appropriate level. In fact, it is possible to extend or replace any subcomponent of the ASP.NET runtime with your own custom-written component. Implementing custom authentication or state services has never been easier.

- **Security.** With built in Windows authentication and per-application configuration, you can be assured that your applications are secure.

Language Support

The Microsoft .NET Platform currently offers built-in support for three languages: C#, Visual Basic, and JScript.

What is ASP.NET Web Forms?

The ASP.NET Web Forms page framework is a scalable common language runtime programming model that can be used on the server to dynamically generate Web pages.

Intended as a logical evolution of ASP (ASP.NET provides syntax compatibility with existing pages), the ASP.NET Web Forms framework

has been specifically designed to address a number of key deficiencies in the previous model. In particular, it provides:

- The ability to create and use reusable UI controls that can encapsulate common functionality and thus reduce the amount of code that a page developer has to write.
- The ability for developers to cleanly structure their page logic in an orderly fashion (not "spaghetti code").
- The ability for development tools to provide strong WYSIWYG design support for pages (existing ASP code is opaque to tools).

ASP.NET Web Forms pages are text files with an .aspx file name extension. They can be deployed throughout an IIS virtual root directory tree. When a browser client requests .aspx resources, the ASP.NET runtime parses and compiles the target file into a .NET Framework class. This class can then be used to dynamically process incoming requests. (Note that the .aspx file is compiled only the first time it is accessed; the compiled type instance is then reused across multiple requests).

An ASP.NET page can be created simply by taking an existing HTML file and changing its file name extension to .aspx (no modification of code is required). For example, the following sample demonstrates a simple HTML page that collects a user's name and category preference and then performs a form postback to the originating page when a button is clicked:

ASP.NET provides syntax compatibility with existing ASP pages. This includes support for <% %> code render blocks that can be in-

termixed with HTML content within an .aspx file. These code blocks execute in a top-down manner at page render time.

Code-Behind Web Forms

ASP.NET supports two methods of authoring dynamic pages. The first is the method shown in the preceding samples, where the page code is physically declared within the originating .aspx file. An alternative approach--known as the code-behind method--enables the page code to be more cleanly separated from the HTML content into an entirely separate file.

Introduction to ASP.NET Server Controls

In addition to (or instead of) using `<% %>` code blocks to program dynamic content, ASP.NET page developers can use ASP.NET server controls to program Web pages. Server controls are declared within an .aspx file using custom tags or intrinsic HTML tags that contain a **runat="server"** attribute value. Intrinsic HTML tags are handled by one of the controls in the **System.Web.UI.HtmlControls** namespace. Any tag that doesn't explicitly map to one of the controls is assigned the type of **System.Web.UI.HtmlControls.HtmlGenericControl**.

Server controls automatically maintain any client-entered values between round trips to the server. This control state is not stored on the server (it is instead stored within an `<input type="hidden">` form field that is round-tripped between requests). Note also that no client-side script is required.

In addition to supporting standard HTML input controls, ASP.NET enables developers to utilize richer custom controls on their pages. For example, the following sample demonstrates how the **<asp:adrotator>** control can be used to dynamically display rotating ads on a page.

1. ASP.NET Web Forms provide an easy and powerful way to build dynamic Web UI.
2. ASP.NET Web Forms pages can target any browser client (there are no script library or cookie requirements).
3. ASP.NET Web Forms pages provide syntax compatibility with existing ASP pages.
4. ASP.NET server controls provide an easy way to encapsulate common functionality.
5. ASP.NET ships with 45 built-in server controls. Developers can also use controls built by third parties.
6. ASP.NET server controls can automatically project both up-level and downlevel HTML.
7. ASP.NET templates provide an easy way to customize the look and feel of list server controls.
8. ASP.NET validation controls provide an easy way to do declarative client or server data validation.

SQL SERVER

DATABASE

A database management, or DBMS, gives the user access to their data and helps them transform the data into information. Such database management systems include dBase, paradox, IMS, Sql Server and SQL Server. These systems allow users to create, update and extract information from their database.

A database is a structured collection of data. Data refers to the characteristics of people, things and events. SQL Server stores each data item in its own fields. In SQL Server, the fields relating to a particular person, thing or event are bundled together to form a single complete unit of data, called a record (it can also be referred to as raw or an occurrence). Each record is made up of a number of fields. No two fields in a record can have the same field name.

During an SQL Server Database design project, the analysis of your business needs identifies all the fields or attributes of interest. If your business needs change over time, you define any additional fields or change the definition of existing fields.

SQL Server Tables

SQL Server stores records relating to each other in a table. Different tables are created for the various groups of information. Related tables are grouped together to form a database.

Primary Key

Every table in SQL Server has a field or a combination of fields that uniquely identifies each record in the table. The Unique identifier is called the Primary Key, or simply the Key. The primary key provides the means to distinguish one record from all other in a table. It allows the user and the database system to identify, locate and refer to one particular record in the database.

Relational Database

Sometimes all the information of interest to a business operation can be stored in one table. SQL Server makes it very easy to link the data in multiple tables. Matching an employee to the department in which they work is one example. This is what makes SQL Server a relational database management system, or RDBMS. It stores data in two or more tables and enables you to define relationships between the table and enables you to define relationships between the tables.

Foreign Key

When a field in one table matches the primary key of another field is referred to as a foreign key. A foreign key is a field or a group

of fields in one table whose values match those of the primary key of another table.

Referential Integrity

Not only does SQL Server allow you to link multiple tables, it also maintains consistency between them. Ensuring that the data among related tables is correctly matched is referred to as maintaining referential integrity.

Data Abstraction

A major purpose of a database system is to provide users with an abstract view of the data. This system hides certain details of how the data is stored and maintained. Data abstraction is divided into three levels.

Physical level: This is the lowest level of abstraction at which one describes how the data are actually stored.

Conceptual Level: At this level of database abstraction all the attributes and what data are actually stored is described and entries and relationship among them.

View level: This is the highest level of abstraction at which one describes only part of the database.

Advantages of RDBMS

- Redundancy can be avoided
- Inconsistency can be eliminated
- Data can be Shared
- Standards can be enforced
- Security restrictions can be applied
- Integrity can be maintained
- Conflicting requirements can be balanced
- Data independence can be achieved.

Disadvantages of DBMS

A significant disadvantage of the DBMS system is cost. In addition to the cost of purchasing or developing the software, the hardware has to be upgraded to allow for the extensive programs and the workspace required for their execution and storage. While centralization reduces duplication, the lack of duplication requires that the database be adequately backed up so that in case of failure the data can be recovered.

FEATURES OF SQL SERVER (RDBMS)

SQL SERVER is one of the leading database management systems (DBMS) because it is the only Database that meets the uncompromising requirements of today's most demanding information systems. From complex decision support systems (DSS) to the most rigorous online transaction processing (OLTP) application, even application that require simultaneous DSS and OLTP access to the same critical data, SQL Server leads the industry in both performance and capability

SQL SERVER is a truly portable, distributed, and open DBMS that delivers unmatched performance, continuous operation and support for every database.

SQL SERVER RDBMS is high performance fault tolerant DBMS which is specially designed for online transactions processing and for handling large database application.

SQL SERVER with transactions processing option offers two features which contribute to very high level of transaction processing throughput, which are

- The row level lock manager

Enterprise wide Data Sharing

The unrivaled portability and connectivity of the SQL SERVER DBMS enables all the systems in the organization to be linked into a singular, integrated computing resource.

Portability

SQL SERVER is fully portable to more than 80 distinct hardware and operating systems platforms, including UNIX, MSDOS, OS/2, Macintosh and dozens of proprietary platforms. This portability gives complete freedom to choose the database server platform that meets the system requirements.

Open Systems

SQL SERVER offers a leading implementation of industry – standard SQL. SQL Server's open architecture integrates SQL SERVER and non –SQL SERVER DBMS with industries most comprehensive collection of tools, application, and third party software products SQL Server's Open architecture provides transparent access to data from other relational database and even non-relational database.

Distributed Data Sharing

SQL Server's networking and distributed database capabilities to access data stored on remote server with the same ease as if the information was stored on a single local computer. A single SQL statement can access data at multiple sites. You can store data where system requirements such as performance, security or availability dictate.

Unmatched Performance

The most advanced architecture in the industry allows the SQL SERVER DBMS to deliver unmatched performance.

Sophisticated Concurrency Control

Real World applications demand access to critical data. With most database Systems application becomes “contention bound” – which performance is limited not by the CPU power or by disk I/O, but user waiting on one another for data access . SQL Server employs full, unrestricted row-level locking and contention free queries to minimize and in many cases entirely eliminates contention wait times.

No I/O Bottlenecks

SQL Server’s fast commit groups commit and deferred write technologies dramatically reduce disk I/O bottlenecks. While some database write whole data block to disk at commit time, SQL Server commits transactions with at most sequential log file on disk at commit time, On high throughput systems, one sequential writes typically group commit multiple transactions. Data read by the transaction remains as shared memory so that other transactions may access that data without reading it again from disk. Since fast commits write all data necessary to the recovery to the log file, modified blocks are written back to the database independently of the transaction commit, when written from memory to disk.

ABOUT AJAX

Ajax is a set of programming techniques or a particular approach to Web programming. These programming techniques involve being able to seamlessly update a Web page or a section of a Web application with input from the server, but without the need for an immediate page refresh. This doesn't mean that the browser doesn't make a connection to the Web server. Indeed, the original article paints a slightly incomplete picture in that it fails to mention that server-side technologies are often still needed. It is very likely that your page, or data from which the page is drawn, must still be updated at some point by a rendezvous with the server. What differs in the Ajax model is that the position at which the page is updated is moved. We'll look at the two models in more detail shortly.

Garrett's article envisaged a world where Web applications could be mirrored Windows applications in their functionality. "Richness," "responsiveness," and "simplicity" were the key words involved. He envisioned a new breed of applications, one that would close the gap between the worlds of Windows and Web applications. He cited Gmail, Google Suggest, and Google Maps as key exponents of this new approach.

The article — and even the term "Ajax" — polarized people. While plenty of people loved it and took up its creed, many developers criticized aspects from the name "Ajax," calling it banal, to the techniques described, which weren't (by any stretch of the imagination) new. There was definitely a hint of the modern art hater's typical criticism about abstract art — "Hey, I could do that and so could my 10-year-old" — about the complaints. Just because people could have been using these techniques to create their Web pages and applications didn't mean they had been. Unfortunately, jealousy and backbiting reigned.

What emerged, though, was a consensus that the techniques and ideas that Jesse James Garrett described really struck a chord (such as "*If we were designing the Web from scratch for applications, we wouldn't make users wait around*" and "*The challenges are for the designers of these applications: to forget what we think we know about the limitations of the Web and begin to imagine a wider, richer range of possibilities*"). It was a call to arms to use existing mature and stable methods to create Web applications rather than the latest flaky beta. It invited developers to leverage the existing knowledge of JavaScript, style

sheets, and the Document Object Model (DOM), instead of sweating blood to get up to speed on the latest tag-based page-building language. It was liberating, and overnight job ads were reworded — "Wanted: developers with five years ~~JavaScript~~ Ajax experience."

Ajax: The Acronym

If you read the Adaptive Path article, then you'll already know that Ajax the acronym stands for *Asynchronous JavaScript and XML*. Here's a curveball: Ajax doesn't have to use XML, and neither does it have to be asynchronous. Ajax applications can use XML, and they can be updated asynchronously. These are quite common tricks and techniques used to update the page, but they are not tied to these technologies.

To reiterate an earlier point, Ajax is "a set of programming techniques," "a particular approach to Web programming." It isn't rigid; it isn't like a members-only club, if you don't use one technique then it isn't Ajax; it's an overall guiding philosophy. How you achieve these objectives on the client is up to you. The objectives, though, prove a good starting point. Jesse James Garrett mentioned in the article "several technologies . . . coming together in powerful new ways." Here are the technologies he specifically mentioned:

- XHTML and CSS
- The Document Object Model (DOM)
- JavaScript
- XML and XSLT
- The XMLHttpRequest object

In reality, to create an application using Ajax techniques you need only three of these: XHTML, the DOM, and JavaScript. If you do any amount of development with Ajax techniques, though, you will almost certainly need to use all of the technologies at some point.

You'll also probably need a server-side language to handle any interaction with the server. This is most typically one of the following three:

- PHP
- ASP.NET (Visual Basic.Net/C#)
- Java

When building a Web page, you'll probably have encountered many or most of these technologies, but perhaps not all, so it's worth having a

quick reminder of what each one is and does, its role in Web development, and how it pertains to Ajax.

XHTML and CSS

You will be familiar with HyperText Markup Language (HTML), the lingua franca of the Web, but perhaps not so familiar with its successor, eXtensible HyperText Markup Language (XHTML). XHTML is the more exacting version of HTML. In fact, it is the HTML standard specified as an XML document. The main difference with this is that whereas HTML has been fairly easygoing and the browser will make a reasonable attempt to display anything you place in tags, XHTML now follows XML's rules. For example, XML documents must be well formed (tags are correctly opened and closed, and nested), and so must XHTML pages. For example, the following is correct nesting:

```
<div>
<h1>
    This is a correctly nested H1 tag
</h1>
</div>
```

The following is incorrect nesting:

```
<div>
<h1>
    This is an incorrectly nested H1 tag
</div>
</h1>
```

Although it might seem to go against the grain of HTML's easygoing and easy-to-code nature, if a page isn't correctly constructed, then you won't be able to perform the kind of Ajax techniques discussed in this article. To use the DOM, the page has to be correctly formed. Otherwise, you won't be able to access the different parts of the page.

Cascading Style Sheets (CSS) are the templates behind HTML pages that describe the presentation and layout of the text and data contained within an HTML page. CSS is of particular interest to the developer because changes made to the style sheet are instantly reflected in the display of the page. The style sheets are linked into the document commonly with the HTML `<link>` tag, although it is possible (but not preferable) to specify style attributes for each individual HTML tag on a page. You can also access CSS properties via the DOM.

In the design of any Web site or Web application, you should make the division between the content/structure of the page and the presentation as clear as possible. Suppose you have 100 pages and you specify the font size on all 100 pages as a style attribute. When you're forced to change the font size you will have to change it on each individual page, instead of changing it just once in the style sheet.

Having a style sheet isn't 100 percent essential, but to keep good organization, style sheets are an indispensable aid.

The Document Object Model (DOM)

The DOM is a representation of the Web page as a hierarchy or tree structure, where every part of the page (the graphics, the text boxes, the buttons, and the text itself) is modeled by the browser.

Before IE 4 and Netscape Navigator 4, not every part of the Web page was accessible to code. In fact, changing text on a Web page had to be done by using server-side technologies or not at all. The whole page was termed a *document*, and that document contained all the HTML tags and text that made up the page. The DOM is specified as a standard by the World Wide Web Consortium, also known as W3C (www.w3.org), and so it is a standard way for all browsers to represent the page. You can pretty much guarantee that when you use JavaScript to alter the background color of a page in IE, it will correctly do so in Mozilla, Safari, or Opera as well. There are exceptions to the rule, though. There are several non-standard methods in IE, and items such as ActiveX controls can't be used in the other browsers.

You can add items to the DOM or alter them using a scripting language (such as JavaScript or VBScript), and they will appear on the page immediately. They are typically addressed in the format that addresses the page in hierarchical format, such as the following code, which addresses a button called "button" on a form and changes the text of that button. Note that in this code fragment, the form element has the name attribute set to form1:

```
document.form1.button.value = "Click Me";
```

Or, you can use methods that can access the specific elements or subsets of elements on the page, such as the `document.getElementById` method, which will return a specific instance of an element that matches the criteria:

```
var myTextBox = document.getElementById("myTextbox");
```

You can then assign values to the variable you have created to alter the values. To make the text box invisible, you could call the following:

```
myTextBox.style.visibility = "visible";
```

Another related method is the `getElementsByTagName` method. The `getElementsByTagName` method will return an array of elements on the Web page of type `NodeList`, all with a given tag name, even if there is only one occurrence of that element on the page. The following code will return all the image elements on the page:

```
var imageElements = document.getElementsByTagName("img");
```

It is also possible to assemble the page by adding new sections to the document known as *nodes*. These can be elements, attributes, or even plain text. For example, you could create a `span` tag that contains a short message and add it to the page as follows:

```
var newTag = document.createElement("span");  
var newText = document.createTextNode("Here is some New Text. Ho  
Hum.");  
newTag.appendChild(newText);  
document.body.appendChild(newTag);
```

All of these DOM techniques are applicable to the client side, and as a result, the browser can update the page or sections of it instantly. Ajax leans on these capabilities very heavily to provide the rich user experience. Also, as mentioned, these techniques have been around since version 4 of IE. It's just that they have been underused. The DOM is an important topic, and it is discussed in much more detail in Chapter 2, "JavaScript Refresher," of the book [Beginning Ajax](#) (Wrox, 2007, ISBN: 978-0-470-10675-4)..

JavaScript

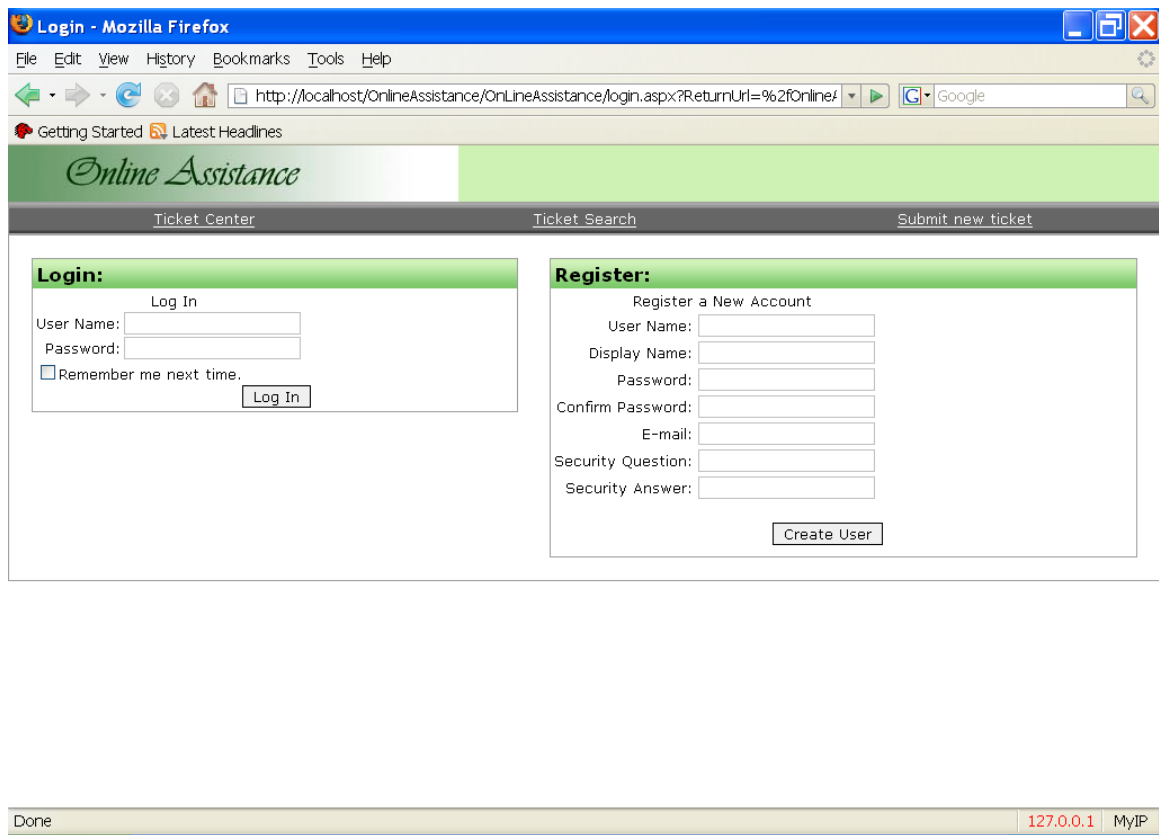
JavaScript is the scripting language of choice of most Web developers. Ajax techniques aren't solely the preserve of JavaScript. VBScript also offers the same capabilities for dynamic updates as well — albeit tied to IE only. While JavaScript has a standard specified in the ECMAScript standard, JavaScript was initially created in Netscape Navigator before such standards existed. Microsoft created its own version of JavaScript (called *JScript*) in parallel, and as a result, each browser's version of

JavaScript is slightly different. Although JavaScript remains a very powerful method for updating your Web pages, some amount of dual-coding is necessary to make sure that the Web pages and applications function in the correct way across browsers. When not possible, some error-handling code will also be necessary.

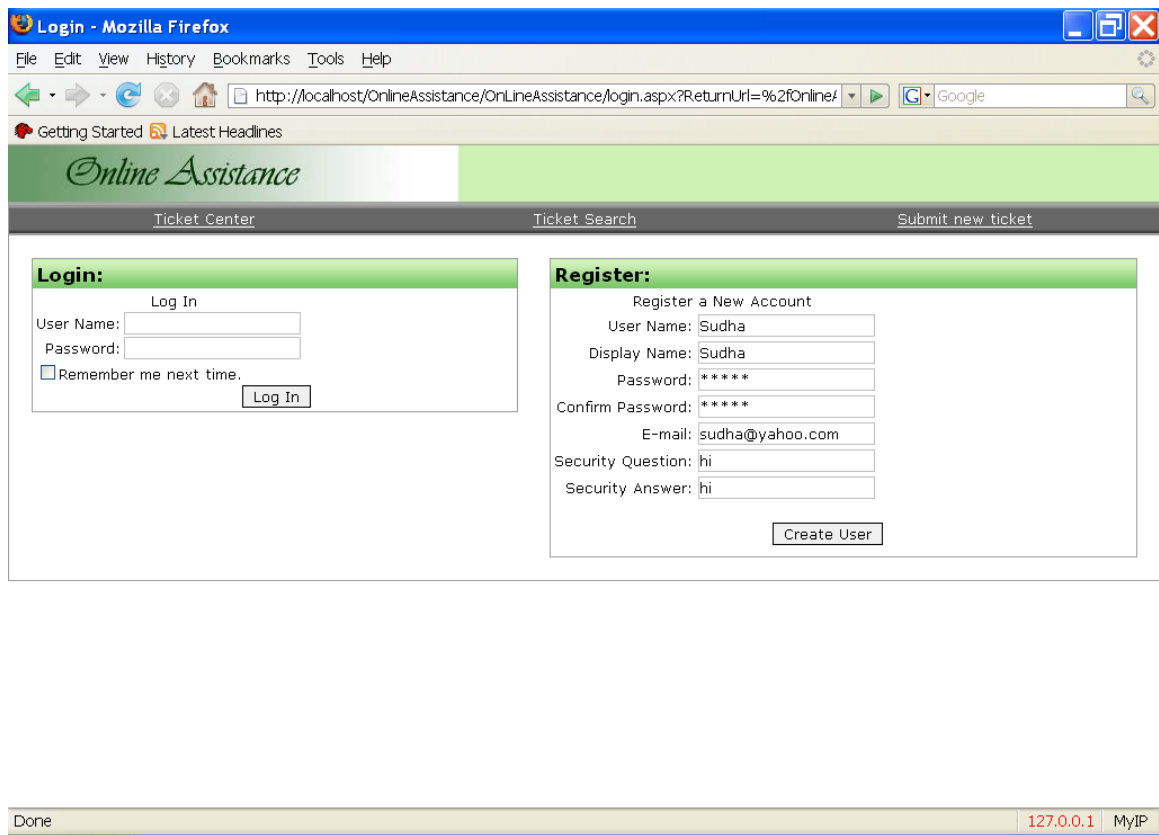
A fair amount of Ajax code will deal with handling cross-browser code and handling errors if and when they arise, unless you can guarantee that your target audience will only ever use one browser (such as on a local intranet). This is an unfortunate set of circumstances that even new versions of IE and Firefox are not able to rectify.

Screens

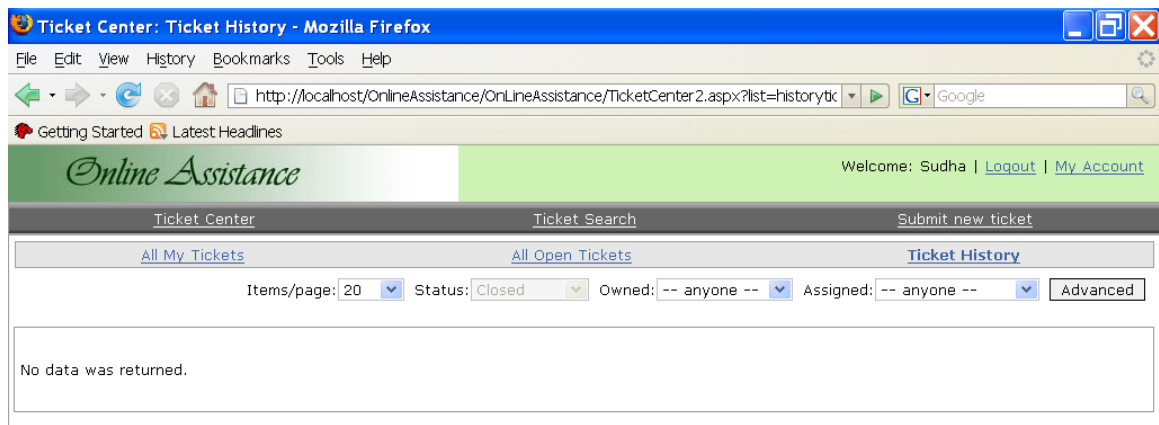
Home Page



Creating New user



Default Page Of the User



All Open Tickets

Ticket Center: All Open Tickets - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/OnlineAssistance/OnLineAssistance/TicketCenter2.aspx?list=open tick Google

Getting Started Latest Headlines

Online Assistance Welcome: Sudha | [Logout](#) | [My Account](#)

Ticket Center Ticket Search Submit new ticket

[All My Tickets](#) [All Open Tickets](#) [Ticket History](#)

Items/page: 20 Status: -- open -- Owned: -- anyone -- Assigned: -- anyone -- [Advanced](#)

ID	Type	Title	Owner	Assigned	Status	Category	Priority	Created	Updated
1	Problem	Test of a toast ticket	admin		Active	Network/Services	Low	04/03/08 12:01 AM	08/20/08 12:24 PM
7	Problem	Configuration Problem	Raghu	admin	Active	Software	High	08/20/08 10:17 AM	08/20/08 12:17 PM
5	Question	Test	admin	admin	Active	Software	High	04/17/08 05:34 PM	08/20/08 11:02 AM
6	Question	Oracle Help	admin	joshi	Active	Software	High	08/01/08 01:16 PM	08/01/08 01:22 PM

First Previous 1 Next Last Page 1 of 1 (4 records)

Ticket Main Page

Online Assistance

Welcome: Sudha | [Logout](#) | [My Account](#)

[Ticket Center](#) [Ticket Search](#) [Submit new ticket](#)

Problem: Configuration Problem

Details:

How to Configure the IIS Server in Windows 2003 Server

Ticket ID: 7

Status: [Active](#)

Priority: High

Category: [Software](#)

Owned by: Raghu

Assigned to: [admin](#)

Affects Customer(s): Yes

Tags:

Created by: Raghu on: 8/20/2008 10:17 AM

Status set by: joshi on: 8/20/2008 12:17 PM

Updated by: joshi on: 8/20/2008 12:17 PM

Add Comment:

[Add Comment](#)

Attachments:

[Browse...](#) [Upload](#)

Activity Log:

Wednesday, 08/20/2008 12:17 PM
joshi assigned the ticket to admin at a priority of High
[I am busy](#)

Wednesday, 08/20/2008 12:17 PM
joshi has re-opened the ticket
[Reopen](#)

Wednesday, 08/20/2008 10:56 AM
Raghu has closed the ticket by force
[Over](#)

Wednesday, 08/20/2008 10:50 AM

To Create new Ticket

Submit New Ticket - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/OnlineAssistance/OnlineAssistance/NewTicket.aspx

Google

Getting Started Latest Headlines

Online Assistance

Welcome: Sudha | [Logout](#) | [My Account](#)

Ticket Center

Ticket Search

Submit new ticket

Create a new ticket

Title:

Type: Category: Priority: Customers:

Problem Software ☐ Has impact on customer(s)

Details:

Create Ticket

You may add file attachments after the ticket has been created.

Done

127.0.0.1 MyIP

Newly created Ticket

Browser address bar: <http://localhost/OnlineAssistance/OnlineAssistance/ViewTicket.aspx?id=8>

Online Assistance

Welcome: Sudha | [Logout](#) | [My Account](#)

Ticket Center Ticket Search Submit new ticket

Problem: Oracle
Details:
How to Configure Oracle Application Server

Ticket ID: 8
Status: [Active](#)
Priority:
Category: [Software](#)
Owned by: Sudha
Assigned to:
Affects Customer(s): No
Tags:
Created by: Sudha on: 8/21/2008 10:22 AM
Status set by: Sudha on: 8/21/2008 10:22 AM
Updated by: Sudha on: 8/21/2008 10:22 AM

Add Comment:

Attachments:

Activity Log:
Thursday, 08/21/2008 10:22 AM
Sudha created the ticket

(8) Problem: Oracle - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/OnlineAssistance/OnLineAssistance/ViewTicket.aspx?id=8

Google

Getting Started Latest Headlines

Online Assistance

Welcome: Sudha | [Logout](#) | [My Account](#)

Ticket Center

Ticket Search

Submit new ticket

Close (force)

Problem: Oracle

Details:

How to Configure Oracle Application Server

Add Comment:

Please give the solution ASAP

Add Comment

Ticket ID: 8

Status: [Active](#)

Priority:

Category: [Software](#)

Owned by: Sudha

Assigned to:

Affects Customer(s): No

Tags:

Created by: Sudha on: 8/21/2008 10:22 AM

Status set by: Sudha on: 8/21/2008 10:22 AM

Updated by: Sudha on: 8/21/2008 10:22 AM

Attachments:

Browse...

Upload

Activity Log:

Thursday, 08/21/2008 10:22 AM

Sudha created the ticket

Done127.0.0.1MyIP

(8) Problem: Oracle - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/OnlineAssistance/OnlineAssistance/ViewTicket.aspx?id=8

Google

Getting Started Latest Headlines

Ticket Center Ticket Search Submit new ticket

Close (force)

Problem: Oracle

Details:

How to Configure Oracle Application Server

Ticket ID: 8

Status: [Active](#)

Priority:

Category: [Software](#)

Owned by: Sudha

Assigned to:

Affects Customer(s): No

Tags:

Created by: Sudha on: 8/21/2008 10:22 AM

Status set by: Sudha on: 8/21/2008 10:22 AM

Updated by: Sudha on: 8/21/2008 10:24 AM

Add Comment:

Add Comment

Attachments:

Browse... Upload

Activity Log:

Thursday, 08/21/2008 10:24 AM

Sudha added comment

Please give the solution ASAP

Thursday, 08/21/2008 10:22 AM

Sudha created the ticket

Done 127.0.0.1 MyIP

(8) Problem: Oracle - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/OnlineAssistance/OnlineAssistance/ViewTicket.aspx?id=8

Google

Getting Started Latest Headlines

Ticket Center Ticket Search Submit new ticket

Close (force)

Problem: Oracle

Details:

How to Configure Oracle Application Server

Add Comment:

Add Comment

Ticket ID: 8

Status: [Active](#)

Priority:

Category: [Software](#)

Owned by: Sudha

Assigned to:

Affects Customer(s): No

Tags:

Created by: Sudha on: 8/21/2008 10:22 AM

Status set by: Sudha on: 8/21/2008 10:22 AM

Updated by: Sudha on: 8/21/2008 10:25 AM

Add Comment:

Add Comment

Attachments:

[YServer.txt](#) - Sudha : 8/21/2008

Browse... Upload

Activity Log:

Thursday, 08/21/2008 10:25 AM

Sudha has added an attachment

New file: YServer.txt

Thursday, 08/21/2008 10:24 AM

Sudha added comment

Done 127.0.0.1 MyIP

Help Desk Login

(8) Problem: Oracle - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/OnlineAssistance/OnLineAssistance/ViewTicket.aspx?id=8 Google

Getting Started Latest Headlines

Online Assistance Welcome: joshi | [Logout](#) | [My Account](#)

[Ticket Center](#) [Ticket Search](#) [Submit new ticket](#)

[Take Over](#) [Assign](#)

Problem: Oracle

Details:

How to Configure Oracle Application Server

Ticket ID: 8

Status: [Active](#)

Priority:

Category: [Software](#)

Owned by: Sudha

Assigned to:

Affects Customer(s): No

Tags:

Created by: Sudha on: 8/21/2008 10:22 AM

Status set by: Sudha on: 8/21/2008 10:22 AM

Updated by: Sudha on: 8/21/2008 10:25 AM

Add Comment:

[Add Comment](#)

Attachments:

[YServer.txt](#) - Sudha : 8/21/2008

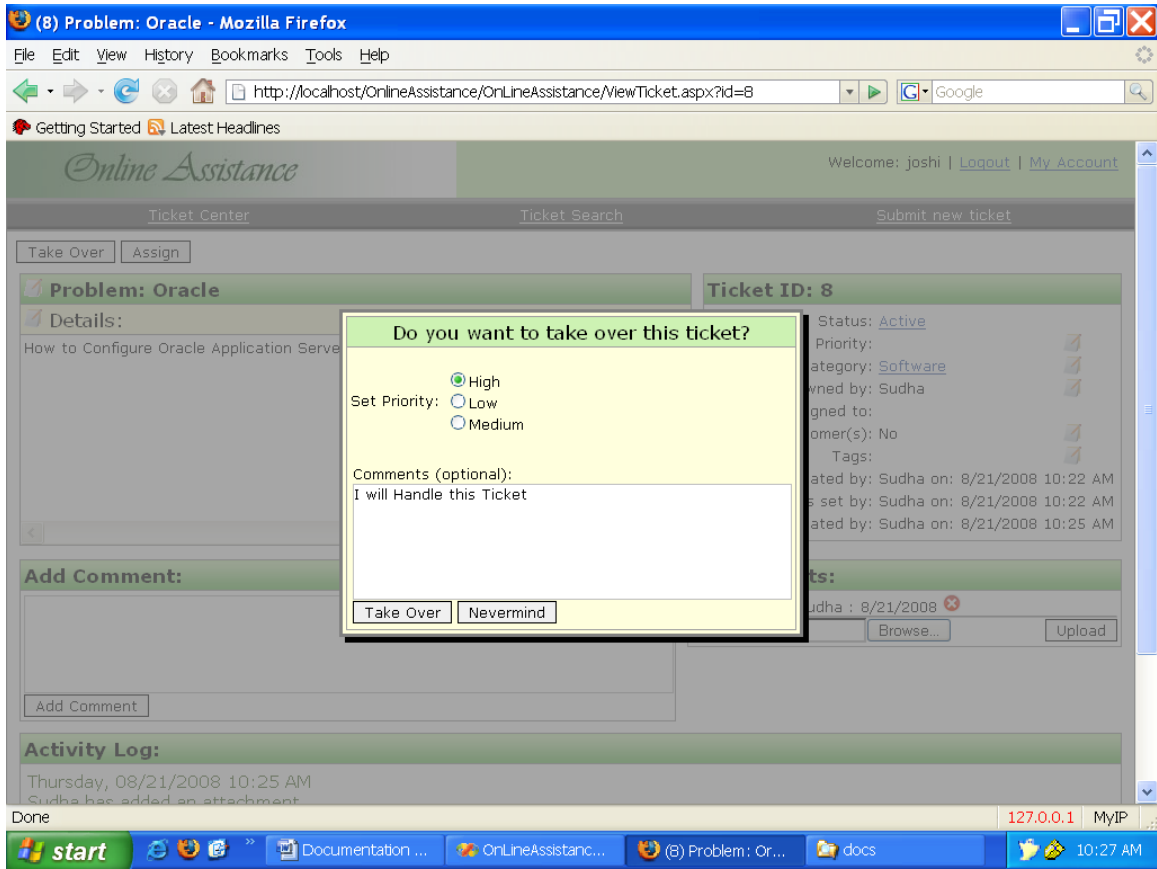
[Browse...](#) [Upload](#)

Activity Log:

Thursday, 08/21/2008 10:25 AM

Sudha has added an attachment

Done 127.0.0.1 MyIP



Online Assistance

Welcome: joshi | [Logout](#) | [My Account](#)

Ticket Center

Ticket Search

Submit new ticket

Resolve

Re-Assign

Request More Info

Give up!

Close (force)

Problem: Oracle

Details:

How to Configure Oracle Application Server

Ticket ID: 8

Status: [Active](#)

Priority: High

Category: [Software](#)

Owned by: Sudha

Assigned to: [joshi](#)

Affects Customer(s): No

Tags:

Created by: Sudha on: 8/21/2008 10:22 AM

Status set by: Sudha on: 8/21/2008 10:22 AM

Updated by: joshi on: 8/21/2008 10:27 AM

Add Comment:

Add Comment

☐ Resolves ticket

Attachments:

[YServer.txt](#) - Sudha : 8/21/2008

Browse...

Upload

Activity Log:

Thursday, 08/21/2008 10:27 AM
joshi has taken over the ticket at a priority of High
I will Handle this Ticket

Thursday, 08/21/2008 10:25 AM
Sudha has added an attachment
New file: YServer.txt

Thursday, 08/21/2008 10:24 AM
Sudha added comment
Please give the solution ASAP

Online Assistance

Welcome: joshi | [Logout](#) | [My Account](#)

Ticket Center Ticket Search Submit new ticket

Resolve Re-Assign Request More Info Give up! Close (force)

Problem: Oracle

Details:

How to Configure Oracle Application Server

Ticket ID: 8

Status: [Active](#)

Priority: [High](#)

Category: [Software](#)

Owned by: [Sudha](#)

Assigned to: [joshi](#)

Customer(s): No

Tags:

Created by: Sudha on: 8/21/2008 10:22 AM

Updated by: Sudha on: 8/21/2008 10:22 AM

Updated by: joshi on: 8/21/2008 10:27 AM

Do you want to request more information from the owner of this ticket?

Comments (required):

Please send the error list u r encountered while installing and configuring the oracle apps server

[Request More Info](#) [Nevermind](#)

Add Comment:

[Add Comment](#) ☐ Resolves ticket

Activity Log:

Thursday, 08/21/2008 10:27 AM

joshi has taken over the ticket at a priority of High

[I will Handle this Ticket](#)

Thursday, 08/21/2008 10:25 AM

Sudha has added an attachment

[New file: YServer.txt](#)

Thursday, 08/21/2008 10:24 AM

Sudha added comment

[Please give the solution ASAP](#)

← → ↻ 🏠 🔍

Online Assistance

Welcome: joshi | [Logout](#) | [My Account](#)

Ticket Center

Ticket Search

Submit new ticket

Resolve

Re-Assign

Cancel More Info

Give up!

Close (force)

Problem: Oracle

Details:

How to Configure Oracle Application Server

Ticket ID: 8

Status: [More Info](#)

Priority: High

Category: [Software](#)

Owned by: Sudha

Assigned to: [joshi](#)

Affects Customer(s): No

Tags:

Created by: Sudha on: 8/21/2008 10:22 AM

Status set by: joshi on: 8/21/2008 10:29 AM

Updated by: joshi on: 8/21/2008 10:29 AM

Add Comment:

Add Comment

☐ Answer request for more information and re-activate ticket

Attachments:

YServer.txt - Sudha : 8/21/2008

Browse...

Upload

Activity Log:

Thursday, 08/21/2008 10:29 AM

joshi has requested more information

Please send the error list u r encountered while installing and configuring the oracle apps server

Thursday, 08/21/2008 10:27 AM

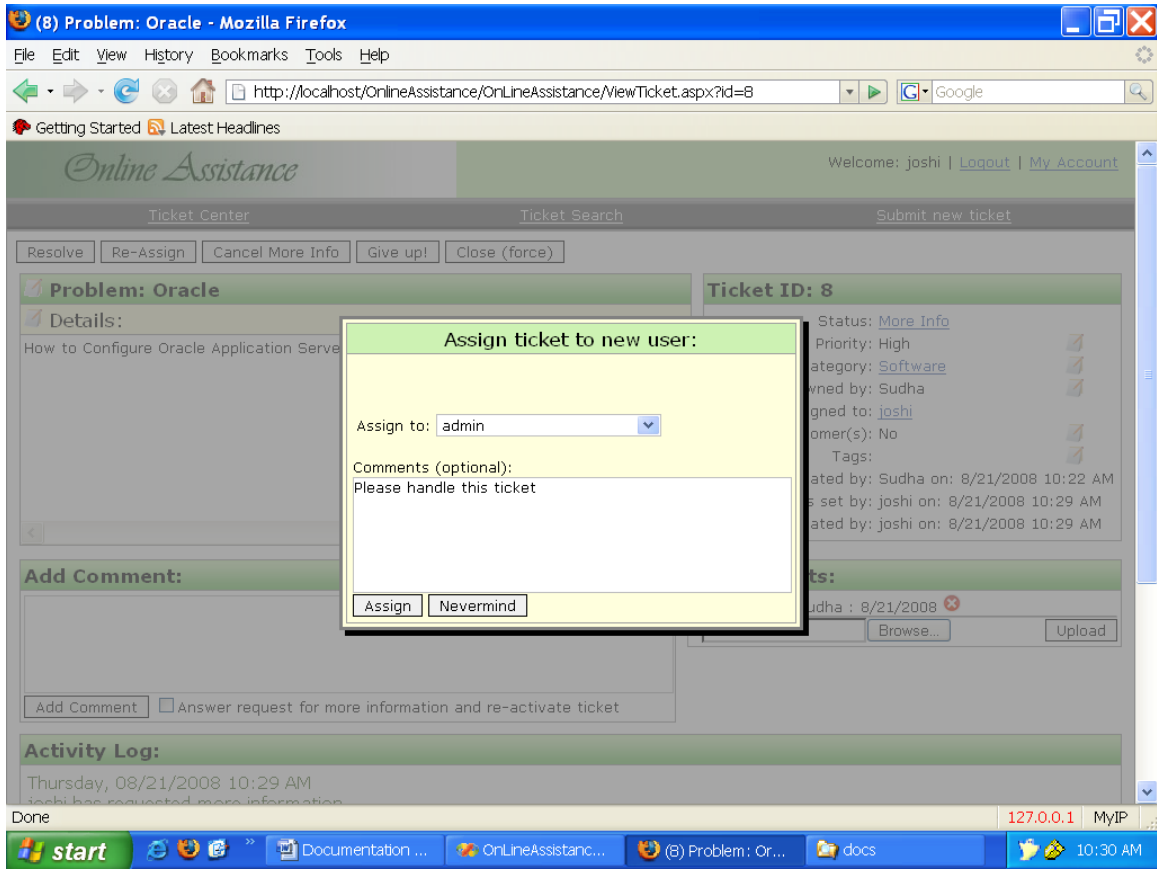
joshi has taken over the ticket at a priority of High

I will Handle this Ticket

Thursday, 08/21/2008 10:25 AM

Sudha has added an attachment

New file: YServer.txt



← → ↻ 🏠 📄 http://localhost/OnlineAssistance/OnLineAssistance/ViewTicket.aspx?id=8 🔍 Google

Online AssistanceWelcome: joshi | [Logout](#) | [My Account](#)

Ticket CenterTicket SearchSubmit new ticket

Take OverRe-Assign

Problem: Oracle

Details:

How to Configure Oracle Application Server

Ticket ID: 8

Status: [More Info](#)

Priority: High

Category: [Software](#)

Owned by: Sudha

Assigned to: [admin](#)

Affects Customer(s): No

Tags:

Created by: Sudha on: 8/21/2008 10:22 AM

Status set by: joshi on: 8/21/2008 10:29 AM

Updated by: joshi on: 8/21/2008 10:31 AM

Add Comment:

Add Comment☐ Answer request for more information and re-activate ticket

Attachments:

[YServer.txt](#) - Sudha : 8/21/2008

Activity Log:

Thursday, 08/21/2008 10:31 AM
joshi passed the ticket to admin
Please handle this ticket

Thursday, 08/21/2008 10:29 AM
joshi has requested more information
Please send the error list u r encountered while installing and configuring the oracle apps server

Thursday, 08/21/2008 10:27 AM
joshi has taken over the ticket at a priority of High
I will Handle this Ticket

Ticket Center: All Open Tickets - Mozilla Firefox

File Edit View History Bookmarks Tools Help

[Getting Started](#) [Latest Headlines](#)

Welcome: joshi | [Logout](#) | [My Account](#)

[Ticket Center](#)
[Ticket Search](#)
[Submit new ticket](#)

[All My Tickets](#)
[All Open Tickets](#)
[Ticket History](#)

Items/page: 20 Status: -- open -- Owned: -- anyone -- Assigned: -- anyone -- [Advanced](#)

ID	Type	Title	Owner	Assigned	Status	Category	Priority	Created	Updated
1	Problem	Test of a toast ticket	admin		Active	Network/Services	Low	04/03/08 12:01 AM	08/20/08 12:24 PM
7	Problem	Configuration Problem	Raghu	admin	Active	Software	High	08/20/08 10:17 AM	08/20/08 12:17 PM
5	Question	Test	admin	admin	Active	Software	High	04/17/08 05:34 PM	08/20/08 11:02 AM
6	Question	Oracle Help	admin	joshi	Active	Software	High	08/01/08 01:16 PM	08/01/08 01:22 PM
8	Problem	Oracle	Sudha	admin	More Info	Software	High	08/21/08 10:22 AM	08/21/08 10:31 AM

[First](#)
[Previous](#)
[1](#)
[Next](#)
[Last](#)
Page 1 of 1 (5 records)

Ticket Center: All Open Tickets - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/OnlineAssistance/OnLineAssistance/TicketCenter2.aspx?list=open tick

Google

Getting Started Latest Headlines

Online Assistance

Welcome: joshi | [Logout](#) | [My Account](#)

Ticket Center

Ticket Search

Submit new ticket

All My Tickets

All Open Tickets

Ticket History

Items/page: 20 Status: -- open -- Owned: admin Assigned: -- anyone --

Advanced

ID	Type	Title	Owner	Assigned	Status	Category	Priority	Created	Updated
1	Problem	Test of a toast ticket	admin		Active	Network/Services	Low	04/03/08 12:01 AM	08/20/08 12:24 PM
5	Question	Test	admin	admin	Active	Software	High	04/17/08 05:34 PM	08/20/08 11:02 AM
6	Question	Oracle Help	admin	joshi	Active	Software	High	08/01/08 01:16 PM	08/01/08 01:22 PM

First Previous 1 Next Last Page 1 of 1 (3 records)

Ticket Center: All Open Tickets - Mozilla Firefox

File Edit View History Bookmarks Tools Help

[http://localhost/OnlineAssistance/OnLineAssistance/TicketCenter2.aspx?list=open tickets](#)

[Getting Started](#)
[Latest Headlines](#)

Online Assistance

Welcome: joshi | [Logout](#) | [My Account](#)

[Ticket Center](#)
[Ticket Search](#)
[Submit new ticket](#)

[All My Tickets](#)
[All Open Tickets](#)
[Ticket History](#)

Items/page: 20
Status: -- open --
Owned: Raghu
Assigned: -- anyone --

ID	Type	Title	Owner	Assigned	Status	Category	Priority	Created	Updated
7	Problem	Configuration Problem	Raghu	admin	Active	Software	High	08/20/08 10:17 AM	08/20/08 12:17 PM

First Previous 1 Next Last
Page 1 of 1 (1 records)

Ticket Center: All Open Tickets - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/OnlineAssistance/OnLineAssistance/TicketCenter2.aspx?list=open tickets Google

Getting Started Latest Headlines

Online Assistance Welcome: joshi | [Logout](#) | [My Account](#)

[Ticket Center](#) [Ticket Search](#) [Submit new ticket](#)

[All My Tickets](#) [All Open Tickets](#) [Ticket History](#)

Items/page: 20 Status: -- open -- Owned: Raghu Assigned: admin [Advanced](#)

ID	Type	Title	Owner	Assigned	Status	Category	Priority	Created	Updated
7	Problem	Configuration Problem	Raghu	admin	Active	Software	High	08/20/08 10:17 AM	08/20/08 12:17 PM

First Previous 1 Next Last Page 1 of 1 (1 records)

Ticket Center: Ticket History - Mozilla Firefox

File Edit View History Bookmarks Tools Help

[http://localhost/OnlineAssistance/OnLineAssistance/TicketCenter2.aspx?list=historytk](#)
Google

Getting Started Latest Headlines

Online Assistance

Welcome: joshi | [Logout](#) | [My Account](#)

Ticket Center

Ticket Search

Submit new ticket

[All My Tickets](#)

[All Open Tickets](#)

[Ticket History](#)

Items/page: 20 Status: Closed Owned: -- anyone -- Assigned: -- anyone -- Advanced

ID	Type	Title	Owner	Assigned	Status	Category	Priority	Created	Updated
4	Request	B Ticket	Toastman	admin	Closed	Hardware	Low	04/11/08 02:21 AM	08/20/08 12:21 PM
3	Problem	A Ticket	admin		Closed	Network/Services	High	04/11/08 02:21 AM	04/14/08 04:09 AM
2	Request	Test ticket 2	admin	joshi	Closed	Non-Technical	High	04/09/08 05:05 PM	08/04/08 10:03 AM

First Previous 1 Next Last

Page 1 of 1 (3 records)

Login - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/OnlineAssistance/OnLineAssistance/login.aspx?ReturnUrl=%2fOnline/ Google

Getting Started Latest Headlines

Online Assistance

[Ticket Center](#) [Ticket Search](#) [Submit new ticket](#)

Login:

Log In

User Name:

Password:

☐ Remember me next time.

Register:

Register a New Account

User Name:

Display Name:

Password:

Confirm Password:

E-mail:

Security Question:

Security Answer:

Ticket Center: Ticket History - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/OnlineAssistance/OnLineAssistance/TicketCenter2.aspx?list=historytic

Google

Getting Started Latest Headlines

Online Assistance

Welcome: admin | [Logout](#) | [My Account](#) | [Administration](#)

Ticket Center Ticket Search Submit new ticket

Unassigned Tickets Tickets Assigned To Me All My Tickets All Open Tickets Ticket History

Items/page: 20 Status: -- any -- Owned: admin Assigned: joshi Advanced

ID	Type	Title	Owner	Assigned	Status	Category	Priority	Created	Updated
6	Question	Oracle Help	admin	joshi	Active	Software	High	08/01/08 01:16 PM	08/01/08 01:22 PM
2	Request	Test ticket 2	admin	joshi	Closed	Non-Technical	High	04/09/08 05:05 PM	08/04/08 10:03 AM

First Previous 1 Next Last Page 1 of 1 (2 records)

Ticket Center: Tickets Assigned To Me - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/OnlineAssistance/OnLineAssistance/TicketCenter2.aspx?list=assigned

Getting Started Latest Headlines

Online Assistance Welcome: admin | [Logout](#) | [My Account](#) | [Administration](#)

Ticket Center Ticket Search Submit new ticket

[Unassigned Tickets](#) **[Tickets Assigned To Me](#)** [All My Tickets](#) [All Open Tickets](#) [Ticket History](#)

Items/page: 20 Status: -- open -- Owned: -- anyone -- Assigned: admin [Advanced](#)

ID	Type	Title	Owner	Assigned	Status	Category	Priority	Created	Updated
7	Problem	Configuration Problem	Raghu	admin	Active	Software	High	08/20/08 10:17 AM	08/20/08 12:17 PM
5	Question	Test	admin	admin	Active	Software	High	04/17/08 05:34 PM	08/20/08 11:02 AM
8	Problem	Oracle	Sudha	admin	More Info	Software	High	08/21/08 10:22 AM	08/21/08 10:31 AM

First Previous 1 Next Last Page 1 of 1 (3 records)

Ticket Center: All My Tickets - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/OnlineAssistance/OnLineAssistance/TicketCenter2.aspx?list=myticket

Getting Started Latest Headlines

Online Assistance Welcome: admin | [Logout](#) | [My Account](#) | [Administration](#)

Ticket Center Ticket Search Submit new ticket

[Unassigned Tickets](#) [Tickets Assigned To Me](#) **[All My Tickets](#)** [All Open Tickets](#) [Ticket History](#)

Items/page: 20 Status: -- open -- Owned: admin Assigned: -- anyone -- [Advanced](#)

ID	Type	Title	Owner	Assigned	Status	Category	Priority	Created	Updated
6	Question	Oracle Help	admin	joshi	Active	Software	High	08/01/08 01:16 PM	08/01/08 01:22 PM
5	Question	Test	admin	admin	Active	Software	High	04/17/08 05:34 PM	08/20/08 11:02 AM
1	Problem	Test of a toast ticket	admin		Active	Network/Services	Low	04/03/08 12:01 AM	08/20/08 12:24 PM

First Previous 1 Next Last Page 1 of 1 (3 records)

Ticket Center: All Open Tickets - Mozilla Firefox

File Edit View History Bookmarks Tools Help

[http://localhost/OnlineAssistance/OnlineAssistance/TicketCenter2.aspx?list=open tickets](#)
Google

Getting Started Latest Headlines

Online Assistance

Welcome: admin | [Logout](#) | [My Account](#) | [Administration](#)

Ticket Center
Ticket Search
Submit new ticket

[Unassigned Tickets](#)
[Tickets Assigned To Me](#)
[All My Tickets](#)
[All Open Tickets](#)
[Ticket History](#)

Items/page: 20 Status: -- open -- Owned: -- anyone -- Assigned: -- anyone -- Advanced

ID	Type	Title	Owner	Assigned	Status	Category	Priority	Created	Updated
1	Problem	Test of a toast ticket	admin		Active	Network/Services	Low	04/03/08 12:01 AM	08/20/08 12:24 PM
7	Problem	Configuration Problem	Raghu	admin	Active	Software	High	08/20/08 10:17 AM	08/20/08 12:17 PM
5	Question	Test	admin	admin	Active	Software	High	04/17/08 05:34 PM	08/20/08 11:02 AM
6	Question	Oracle Help	admin	joshi	Active	Software	High	08/01/08 01:16 PM	08/01/08 01:22 PM
8	Problem	Oracle	Sudha	admin	More Info	Software	High	08/21/08 10:22 AM	08/21/08 10:31 AM

First Previous 1 Next Last Page 1 of 1 (5 records)

(2) Request: Test ticket 2 - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/OnlineAssistance/OnLineAssistance/ViewTicket.aspx?id=2

Google

Getting Started Latest Headlines

Online Assistance

Welcome: admin | [Logout](#) | [My Account](#) | [Administration](#)

Ticket Center Ticket Search Submit new ticket

Re-Open

Request: Test ticket 2

Details:

Test ticket 2

Ticket ID: 2

Status: [Closed](#)
Priority: High
Category: [Non-Technical](#)
Owned by: admin
Assigned to: [joshi](#)
Affects Customer(s): No
Tags:
Created by: admin on: 4/9/2008 5:05 PM
Status set by: joshi on: 8/4/2008 10:03 AM
Updated by: joshi on: 8/4/2008 10:03 AM

Attachments:

Activity Log:

Monday, 08/04/2008 10:03 AM
joshi has closed the ticket by force
Over

Monday, 08/04/2008 09:56 AM
admin assigned the ticket to joshi without comment

Wednesday, 04/09/2008 17:05 PM
admin created the ticket

Done127.0.0.1MyIP

Ticket Center: Unassigned Tickets - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/OnlineAssistance/OnLineAssistance/TicketCenter2.aspx

Getting Started Latest Headlines

Online Assistance

Welcome: admin | [Logout](#) | [My Account](#) | [Administration](#)

Ticket Center Ticket Search Submit new ticket

Unassigned Tickets Tickets Assigned To Me All My Tickets All Open Tickets Ticket History

Items/page: 50 Status: -- open -- Owned: -- anyone -- Assigned: -- unassigned -- Advanced

Advanced Settings

Advanced Sorting

Assigned To

Add

Ticket Id

Sort Descending

Done

Ticket Id

Type

Category

Title

Created By

Created Date

Owner

Current Status

Last Update By

Last Update Date

Priority

Affects Customer

ID	Type	Owner	Assigned	Status	Category	Priority	Created	Updated
1	Problem	admin		Active	Network/Services	Low	04/03/08 12:01 AM	08/20/08 12:24 PM

First Previous 1 Next Last Page 1 of 1 (1 records)

Done

127.0.0.1 MyIP

start Documentation ... OnLineAssistanc... Ticket Center: U... docs 10:38 AM

Ticket Search - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/OnlineAssistance/OnLineAssistance/TicketSearch.aspx

Getting Started Latest Headlines

Online Assistance Welcome: admin | [Logout](#) | [My Account](#) | [Administration](#)

Ticket Center **Ticket Search** Submit new ticket

Select your search options:

Search For: oracle

☒ Search Title ☐ Search Details ☐ Search Tags

Status: Any Status

Assigned To: Anyone

Category: Any Category

Type: Any Type

Owned By: Anyone

Done 127.0.0.1 MyIP

start Documentation ... OnLineAssistanc... Ticket Search - ... docs

10:39 AM

Ticket Search - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/OnlineAssistance/OnLineAssistance/TicketSearch.aspx

Google

Getting Started Latest Headlines

Online Assistance

Welcome: admin | [Logout](#) | [My Account](#) | [Administration](#)

Ticket Center

Ticket Search

[Submit new ticket](#)

Select your search options:

Search For:
oracle

☒ Search Title ☐ Search Details ☐ Search Tags

Status: Any Status

Assigned To: Anyone

Category: Any Category

Type: Any Type

Owned By: Anyone

ID	Type	Title	Owner	Assigned	Status	Category	Priority	Created	Updated
8	Problem	Oracle	Sudha	admin	More Info	Software	High	08/21/08 10:22 AM	08/21/08 10:31 AM
6	Question	Oracle Help	admin	joshi	Active	Software	High	08/01/08 01:16 PM	08/01/08 01:22 PM

First Previous 1 Next Last Page 1 of 1 (2 records)

Ticket Search - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/OnlineAssistance/OnLineAssistance/TicketSearch.aspx

Getting Started Latest Headlines

Online Assistance Welcome: admin | [Logout](#) | [My Account](#) | [Administration](#)

Ticket Center **Ticket Search** Submit new ticket

Select your search options:

Search For: oracle

☒ Search Title ☐ Search Details ☐ Search Tags

Status: Any Status

Assigned To: Anyone

Category: Any Status

Type: Not Closed
Only Active
Only More Info
Only Resolved
Only Closed

Owened By: Anyone

ID	Type	Title	Owner	Assigned To	Status	Category	Priority	Created	Updated
8	Problem	Oracle	Sudha	admin	More Info	Software	High	08/21/08 10:22 AM	08/21/08 10:31 AM
6	Question	Oracle Help	admin	joshi	Active	Software	High	08/01/08 01:16 PM	08/01/08 01:22 PM

First Previous 1 Next Last Page 1 of 1 (2 records)

Done 127.0.0.1 MyIP

start Documentation ... OnLineAssistanc... Ticket Search - ... docs 10:40 AM

Ticket Search - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/OnlineAssistance/OnlineAssistance/TicketSearch.aspx

Getting Started Latest Headlines

Online Assistance Welcome: admin | [Logout](#) | [My Account](#) | [Administration](#)

Ticket Center **Ticket Search** [Submit new ticket](#)

Select your search options:

Search For: oracle

☒ Search Title ☐ Search Details ☐ Search Tags

Status: Any Status

Assigned To: Anyone

Category: Any Category

Type: Any Category
Hardware
Software
Network/Services
Non-Technical

Owened By: Anyone

ID	Type	Title	Owner	Assign	Category	Priority	Created	Updated
8	Problem	Oracle	Sudha	admin	More Info Software	High	08/21/08 10:22 AM	08/21/08 10:31 AM
6	Question	Oracle Help	admin	joshi	Active Software	High	08/01/08 01:16 PM	08/01/08 01:22 PM

First Previous 1 Next Last Page 1 of 1 (2 records)

Done

127.0.0.1 MyIP

start Documentation ... OnLineAssistanc... Ticket Search - ... docs

10:40 AM

Ticket Search - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/OnlineAssistance/OnLineAssistance/TicketSearch.aspx

Getting Started Latest Headlines

Online Assistance Welcome: admin | [Logout](#) | [My Account](#) | [Administration](#)

Ticket Center **Ticket Search** [Submit new ticket](#)

Select your search options:

Search For: oracle

☒ Search Title ☐ Search Details ☐ Search Tags

Status: Any Status

Assigned To: Anyone

Category: Any Category

Type: Any Type

Owned By: Anyone

ID	Type	Title	Owner	Assigned	Status	Category	Priority	Created	Updated
8	Problem	Oracle	Sudha	admin	More Info	Software	High	08/21/08 10:22 AM	08/21/08 10:31 AM
6	Question	Oracle Help	admin	joshi	Active	Software	High	08/01/08 01:16 PM	08/01/08 01:22 PM

First Previous 1 Next Last Page 1 of 1 (2 records)

Done 127.0.0.1 MyIP

start Documentation ... OnLineAssistanc... Ticket Search - ... docs 10:40 AM

Ticket Search - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/OnlineAssistance/OnLineAssistance/TicketSearch.aspx

Getting Started Latest Headlines

Online Assistance Welcome: admin | [Logout](#) | [My Account](#) | [Administration](#)

Ticket Center **Ticket Search** Submit new ticket

Select your search options:

Search For: oracle

☒ Search Title ☐ Search Details ☐ Search Tags

Status: Any Status Assigned To: Anyone

Category: Any Category Type: Any Type Owned By: Anyone

Type to search

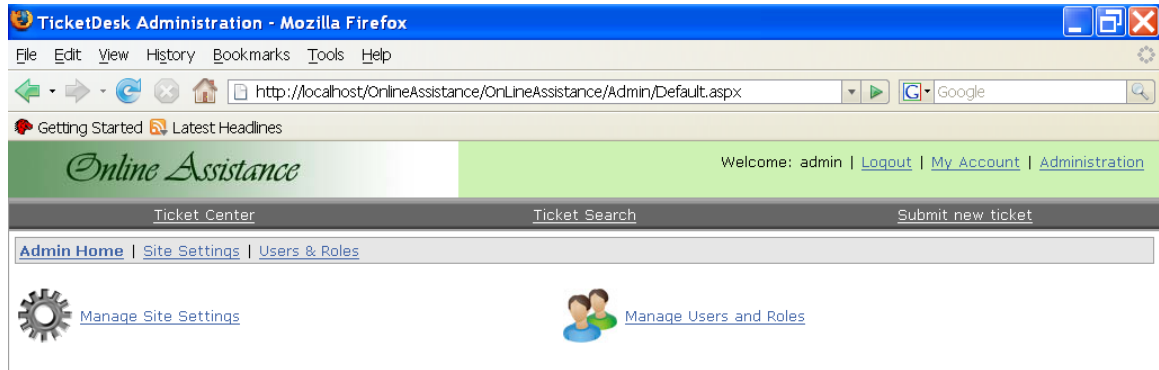
ID	Type	Title	Owner	Assigned	Status	Category	Priority	Updated
8	Problem	Oracle	Sudha	admin	More Info	Software	High	08/21/08 10:31 AM
6	Question	Oracle Help	admin	joshi	Active	Software	High	08/01/08 01:16 PM 08/01/08 01:22 PM

First Previous 1 Next Last Page 1 of 1 (2 records)

Done 127.0.0.1 MyIP

start Documentation ... OnLineAssistanc... Ticket Search - ... docs 10:41 AM

Administration Main Page



TicketDesk Site Settings - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/OnlineAssistance/OnLineAssistance/Admin/Settings.aspx

Getting Started Latest Headlines

Online Assistance

Welcome: admin | [Logout](#) | [My Account](#) | [Administration](#)

[Ticket Center](#) [Ticket Search](#) [Submit new ticket](#)

[Admin Home](#) | [Site Settings](#) | [Users & Roles](#)

Priorities

High

Low

Medium

Add

- Drag-and-drop to change the display order of Priorities.
- To merge two priorities, rename one to the same name as the other. The system will manage the merge automatically and update tickets using the old priority with the merged priority.
- You cannot delete a priority as that would leave tickets without an associates priority. Instead, merge the priority with an existing priority.
- Renaming or merging a priority will update all existing tickets. Comments will be added to modified tickets, but notifications will **not** be sent.

Categories

Ticket Types

Done

127.0.0.1 MyIP

start Documentation ... OnLineAssistanc... TicketDesk Site ... docs

10:42 AM

TicketDesk Site Settings - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/OnlineAssistance/OnLineAssistance/Admin/Settings.aspx

Google

Getting Started Latest Headlines

Online Assistance

Welcome: admin | [Logout](#) | [My Account](#) | [Administration](#)

Ticket Center

Ticket Search

Submit new ticket

[Admin Home](#) | [Site Settings](#) | [Users & Roles](#)

Priorities

High

Low

Medium

Average

Add

- Drag-and-drop to change the display order of Priorities.
- To merge two priorities, rename one to the same name as the other. The system will manage the merge automatically and update tickets using the old priority with the merged priority.
- You cannot delete a priority as that would leave tickets without an associates priority. Instead, merge the priority with an existing priority.
- Renaming or merging a priority will update all existing tickets. Comments will be added to modified tickets, but notifications will **not** be sent.

Categories

Ticket Types

Done

127.0.0.1 MyIP

TicketDesk Site Settings - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/OnlineAssistance/OnlineAssistance/Admin/Settings.aspx

Getting Started Latest Headlines

Online Assistance Welcome: admin | [Logout](#) | [My Account](#) | [Administration](#)

[Ticket Center](#) [Ticket Search](#) [Submit new ticket](#)

[Admin Home](#) | [Site Settings](#) | [Users & Roles](#)

Priorities

Categories

Hardware

Software

Network/Services

Non-Technical

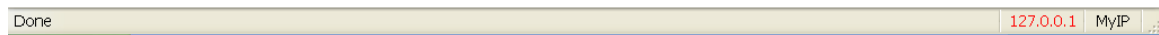
Add

- Drag-and-drop to change the display order of categories.
- To merge two categories, rename one to the same name as the other. The system will manage the merge automatically and update tickets using the old category with the merged category.
- You cannot delete a category as that would leave tickets without an associates category. Instead, merge the category with an existing category.
- Renaming or merging a category will update all existing tickets. Comments will be added to modified tickets, but notifications will **not** be sent.

Ticket Types

Done 127.0.0.1 MyIP

start Documentation ... OnLineAssistanc... TicketDesk Site ... docs 10:43 AM



TicketDesk Site Settings - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/OnlineAssistance/OnlineAssistance/Admin/Settings.aspx

Getting Started Latest Headlines

Online Assistance Welcome: admin | [Logout](#) | [My Account](#) | [Administration](#)


[Ticket Center](#) [Ticket Search](#) [Submit new ticket](#)


[Admin Home](#) | [Site Settings](#) | [Users & Roles](#)


Priorities

Categories

Ticket Types

 Question

 Problem

 Request

- Drag-and-drop to change the display order of types.
- To merge two types, rename one to the same name as the other. The system will manage the merge automatically and update tickets using the old type with the merged type.
- You cannot delete a type as that would leave tickets without an associates type. Instead, merge the type with an existing type.
- Renaming or merging a Type will update all existing tickets. Comments will be added to modified tickets, but notifications will **not** be sent.

Done 127.0.0.1 MyIP

start Documentation ... OnLineAssistanc... TicketDesk Site ... docs 10:44 AM

TicketDesk Site Settings - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/OnlineAssistance/OnlineAssistance/Admin/Settings.aspx

Getting Started Latest Headlines

Online Assistance Welcome: admin | [Logout](#) | [My Account](#) | [Administration](#)

[Ticket Center](#) [Ticket Search](#) [Submit new ticket](#)

[Admin Home](#) | [Site Settings](#) | [Users & Roles](#)

Priorities

Categories

Ticket Types

☐ Question

☐ Problem

☐ Request

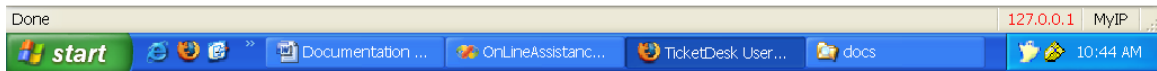
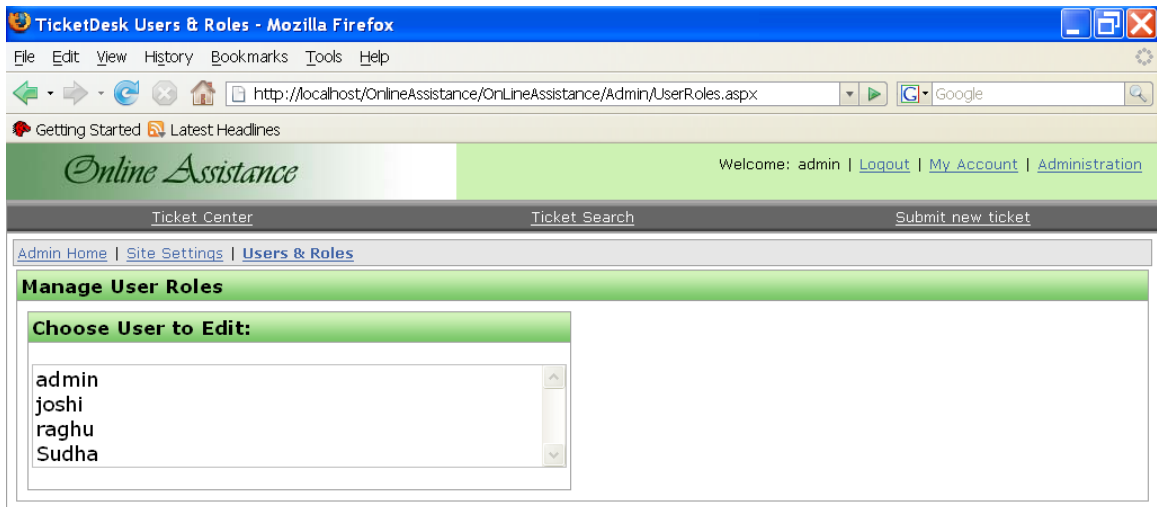
☐ Help

Add

- Drag-and-drop to change the display order of types.
- To merge two types, rename one to the same name as the other. The system will manage the merge automatically and update tickets using the old type with the merged type.
- You cannot delete a type as that would leave tickets without an associates type. Instead, merge the type with an existing type.
- Renaming or merging a Type will update all existing tickets. Comments will be added to modified tickets, but notifications will **not** be sent.

Done 127.0.0.1 MyIP

start Documentation ... OnLineAssistanc... TicketDesk Site ... docs 10:44 AM



TicketDesk Users & Roles - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/OnlineAssistance/OnLineAssistance/Admin/UserRoles.aspx

Getting Started Latest Headlines

Online Assistance Welcome: admin | [Logout](#) | [My Account](#) | [Administration](#)

Ticket Center Ticket Search Submit new ticket

[Admin Home](#) | [Site Settings](#) | [Users & Roles](#)

Manage User Roles

Choose User to Edit:
admin
joshi
raghu
Sudha

Choose Roles for User:
☒ TicketSubmitters
☒ HelpDesk
☒ Administrators
[Delete User](#)

Done 127.0.0.1 MyIP

start Documentation ... OnLineAssistanc... TicketDesk User... docs 10:45 AM

TicketDesk Users & Roles - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/OnlineAssistance/OnLineAssistance/Admin/UserRoles.aspx

Google

Getting Started Latest Headlines

Online Assistance

Welcome: admin | [Logout](#) | [My Account](#) | [Administration](#)

Ticket Center

Ticket Search

Submit new ticket

[Admin Home](#) | [Site Settings](#) | [Users & Roles](#)

Manage User Roles

Choose User to Edit:

adminjoshi
raghu
Sudha

Choose Roles for User:

☐ TicketSubmitters

☒ HelpDesk

☒ Administrators

Delete User

Roles Changed!

TicketDesk Users & Roles - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/OnlineAssistance/OnLineAssistance/Admin/UserRoles.aspx

Getting Started Latest Headlines

Online Assistance Welcome: admin | [Logout](#) | [My Account](#) | [Administration](#)

Ticket Center Ticket Search Submit new ticket

[Admin Home](#) | [Site Settings](#) | [Users & Roles](#)

Manage User Roles

Choose User to Edit:
admin
joshi
raghu
Sudha

Choose Roles for User:
☐ TicketSubmitters
☒ HelpDesk
☒ Administrators

Roles Changed!

Done 127.0.0.1 MyIP

start Documentation ... OnLineAssistanc... TicketDesk User... docs 10:45 AM

TicketDesk Users & Roles - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/OnlineAssistance/OnLineAssistance/Admin/UserRoles.aspx

Getting Started Latest Headlines

Online Assistance Welcome: admin | [Logout](#) | [My Account](#) | [Administration](#)

Ticket Center Ticket Search Submit new ticket

[Admin Home](#) | [Site Settings](#) | [Users & Roles](#)

Manage User Roles

Choose User to Edit:
admin
joshi
raghu
Sudha

Choose Roles for User:
☒ TicketSubmitters
☐ HelpDesk
☐ Administrators

The page at http://localhost says:

Are you sure you would like to delete this user (this action cannot be undone)?

OK Cancel

Done 127.0.0.1 MyIP

start Documentation ... OnLineAssistanc... TicketDesk User... docs 10:46 AM

TicketDesk Users & Roles - Mozilla Firefox

File Edit View History Bookmarks Tools Help

http://localhost/OnlineAssistance/OnLineAssistance/Admin/UserRoles.aspx

Getting Started Latest Headlines

Online Assistance Welcome: admin | [Logout](#) | [My Account](#) | [Administration](#)

Ticket Center Ticket Search Submit new ticket

[Admin Home](#) | [Site Settings](#) | [Users & Roles](#)

Manage User Roles

Choose User to Edit:

admin
joshi
raghu

User 'Sudha' has been Deleted

Done 127.0.0.1 MyIP

start Documentation ... OnLineAssistanc... TicketDesk User... docs 10:46 AM

9. TESTING AND IMPLEMENTATION

Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved. System testing is its utility as user -oriented vehicle before implementation. The first test of a system is to see whether it produces the correct outputs. The purpose of system testing is to identify and correct errors in the candidate system. System Testing consists of Program testing, string testing, system testing, System documentation, user acceptance testing.

One of the important testing in this project is Black box testing. Black box testing attempts to find errors in the following categories,

Missing functions, interface errors, errors in the data structures, performance errors and initialization and termination errors.

Black box testing is sometimes called behavioral testing. Black box testing broadens our focus and called as "testing in the large".

Black box testing focuses on the information domains of the software deriving test cases by partitioning the input and output domain of a program in a manner that provides through test coverage Black box testing consists of Graph based testing methods, Equivalence partitioning, Boundary Value Analysis and Comparison Testing.

Graph Based Testing methods explore the relationships between and behavior of program objects. Equivalence partitioning divides the input domain into classes of data that are likely to exercise specific software functions. Boundary Value Analysis probes the program's ability to handle data at the limits of acceptability.

Testing never ends, it just gets transferred from you to your customer. Every time customer uses the program, a test is being conducted. By applying test case design we can achieve more complete testing and thereby uncover and correct the highest number of errors before the test begins.

Testing

1) COMPILATION TEST:

- It was a good idea to do our stress testing early on, because it gave us time to fix some of the unexpected deadlocks and stability problems that only occurred when components were exposed to very high transaction volumes.

2) EXECUTION TEST:

- This program was successfully loaded and executed. Because of good programming there was no execution error.

3) OUTPUT TEST:

- The successful output screens are placed in the output screens section.

10. CONCLUSION

FUTURE ENHANCEMENT

1. Provide more user friendly environment.
2. Implement the Auto display of the solutions from the database whenever the similar ticket is submitted.
3. Implement the chatting module between helpdesk and ticket submitters.

CONCLUSION

- The project has been appreciated by all the users in the organization and by client.
- It is easy to use, since it uses the **GUI** provided in the user dialog.
- User friendly screens are provided.
- The usage of software increases the efficiency, decreases the effort.
- It has been efficiently employed as a Online Assistance
- It has been thoroughly tested and implemented.

BIBLIOGRAPHY

SOFTWARE ENGINEERING

By Roger.S. Pressman

SQL FOR PROFESSIONALS

By Jain

C#.NET Black Book

By Evangeleous Petereous

Professional ASP.NET 3.5

By Wrox

MSDN 2008

By Microsoft

